THE UNIVERSITY OF ALABAMA

COLLEGE OF ENGINEERING
BUREAU OF ENGINEERING RESEARCH

FINAL REPORT

on

Contract NAS8-20172

A FAST-INITIALIZING DIGITAL EQUALIZER WITH ON-LINE TRACKING FOR DATA COMMUNICATIONS

by

RONALD C. HOUTS
Project Director

and

WILLIAM J. BARKSDALE
Research Associate

Prepared for

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

May 1974

BER Report No. 166-102

A FAST-INITIALIZING DIGITAL EQUALIZER WITH ON-LINE TRACKING FOR DATA COMMUNICATIONS

by

RONALD C. HOUTS
Project Director

and

WILLIAM J. BARKSDALE
Research Associate

MAY, 1974

TECHNICAL REPORT NUMBER 166-102

Prepared for

National Aeronautics and Space Administration
Marshall Space Flight Center
Huntsville, Alabama

Under Contract Number

NAS8-20172

Communication Systems Group
Bureau of Engineering Research
The University of Alabama

ABSTRACT

A theory is developed for a digital equalizer for use in reducing intersymbol interference (ISI) on high-speed data communications channels. The equalizer is initialized with a single isolated transmitter pulse, provided the signal-to-noise ratio (SNR) is not unusually low, then switches to a decision-directed, on-line mode of operation that allows tracking of channel variations. Conditions for optimal tap-gain settings are obtained first for a transversal equalizer structure by using a mean-squared error (MSE) criterion, a first-order gradient algorithm to determine the adjustable equalizer tap-gains, and a sequence of isolated initializing pulses. Since the rate of tap-gain convergence depends on the eigenvalues of a channel-output correlation matrix $\underline{\tilde{A}}$ convergence can be improved by making a linear transformation on $\underline{\tilde{A}}$ to obtain a new correlation matrix $\underline{A}$. The corresponding change in equalizer structure results in a set of parallel transversal filter-sections, the weighted outputs of which are summed to produce the equalizer output. It is shown that by properly selecting the tap-gains of each transversal filter-section, the equalizer can be adjusted to minimize the MSE on the first iteration of the algorithm. A method

is proposed for making this selection using the pseudoinverse of $\underline{A}$ to obtain a "best" approximation to an orthogonality condition. By using a Hadamard matrix it is shown that $\underline{A}$ is similar to a projection matrix. Thus all eigenvalues are either 0 to 1, which is the desired condition for initialization in one algorithm iteration.

When the channel SNR is low initialization can be improved by averaging several initializing pulses in order to better identify the channel. Next a simple approximation to the error criterion is used to obtain an on-line algorithm for noisy channels; it makes small adjustments to the filter-section output weights to compensate for channel noise and time-variations, as well as poor initialization. Convergence of this on-line algorithm is shown in a statistical sense under a stated hypothesis, and suggestions are made for detection and correction of poor channel tracking. Finally the equalizer design is evaluated by means of a digital computer simulation and the results are presented and discussed for three baseband channel models. The interaction among several key equalizer parameters is studied for a wide range of SNR.

LIST OF SYMBOLS

Symbol	Meaning
$\underline{v}$	a column vector
$\underline{M}$	a matrix
δ_{mn}	the Kronecker delta
$\underline{M}^T$	transpose of $\underline{M}$
$\underline{M}^{-1}$	inverse of $\underline{M}$
$\underline{M}^{-I}$	pseudoinverse of $\underline{M}$
$\underline{M}^*$	complex conjugate of $\underline{M}$
$\underline{M}^H$	complex conjugate (hermitian) transpose of $\underline{M}$
diag(a,b,...c)	diagonal matrix with diagonal elements a,b,...c
det($\underline{M}$)	determinant of $\underline{M}$
$\underline{I}_N$	N x N identity matrix
$\underline{O}_N$	N x N null matrix
∇	gradient operator
E	mathematical expectation operator
$\langle \underline{x}, \underline{y} \rangle$	inner (scalar) product of $\underline{x}$ and $\underline{y}$
$\|\| \underline{x} \|\|$	norm of vector $\underline{x}$
mod n	modulo n
(M,N,D)	set of parameters describing the equalizer
j	$\sqrt{-1}$
s	second (of time)

ACKNOWLEDGMENT

The authors would like to express their appreciation to the Telemetry and Data Technology Branch of the Astrionics Laboratory, Marshall Space Flight Center, for the support of this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Motivated by the need to transmit binary data directly between computers, there has been a steady increase in the use of digital techniques for data transmission. This is due in large part to the profound advances in solid-state technology which have resulted in small, reliable, fast, and inexpensive digital components. Attempts to increase the transmission rate of digital data over band-limited channels are often limited by interference among successive data symbols, an effect known as intersymbol interference (ISI). The reduction of ISI by signal processing techniques is known as equalization.

1.1. THE EQUALIZATION OF DIGITAL CHANNELS

The basic principles of digital communications are well documented in standard texts [1]. Such systems always contain unwanted signals which are usually classified collectively as noise. For this study it is convenient to divide noise into two subcategories; background noise caused by external sources such as thermal excitation and ISI caused by the overlap in time of the received signals. As the data transmission rate is increased for a fixed channel

bandwidth, the ISI becomes increasingly important for the following reason. Over the nominal channel passband all practical channels have nonlinear phase vs. frequency characteristics which cause different frequency components of a transmitted signal to be delayed by different amounts. This results in a transmitted pulse being spread out, or dispersed, in time. This time dispersion is further aggravated by the fact that all physical channels are bandlimited and thus attenuate the high frequency components of transmitted signals. At the channel output these successive time dispersed signals overlap to give a composite signal equal to their algebraic sum. Fig. 1.1 shows such a situation, with the circled points indicating those values measured by a receiver that takes a single sample per bit. In the figure time has been normalized, and T_d represents the normalized time delay of the transmitted pulse caused by the channel. The channel output sample at $t = T_d + 2$, for example, is closer to unity than zero; therefore, the zero transmitted at $t = 2$ would be mistaken for a one, resulting in a message error. Errors can also be caused by background noise or time variation of channel parameters. Some ways to reduce the error rate include raising the signal-to-noise ratio (SNR), introducing redundancy through coding, and the use of pulse-shaping filters to reshape the transmitted or received signals. This latter approach is called channel equalization, and the filter is termed an equalizer.

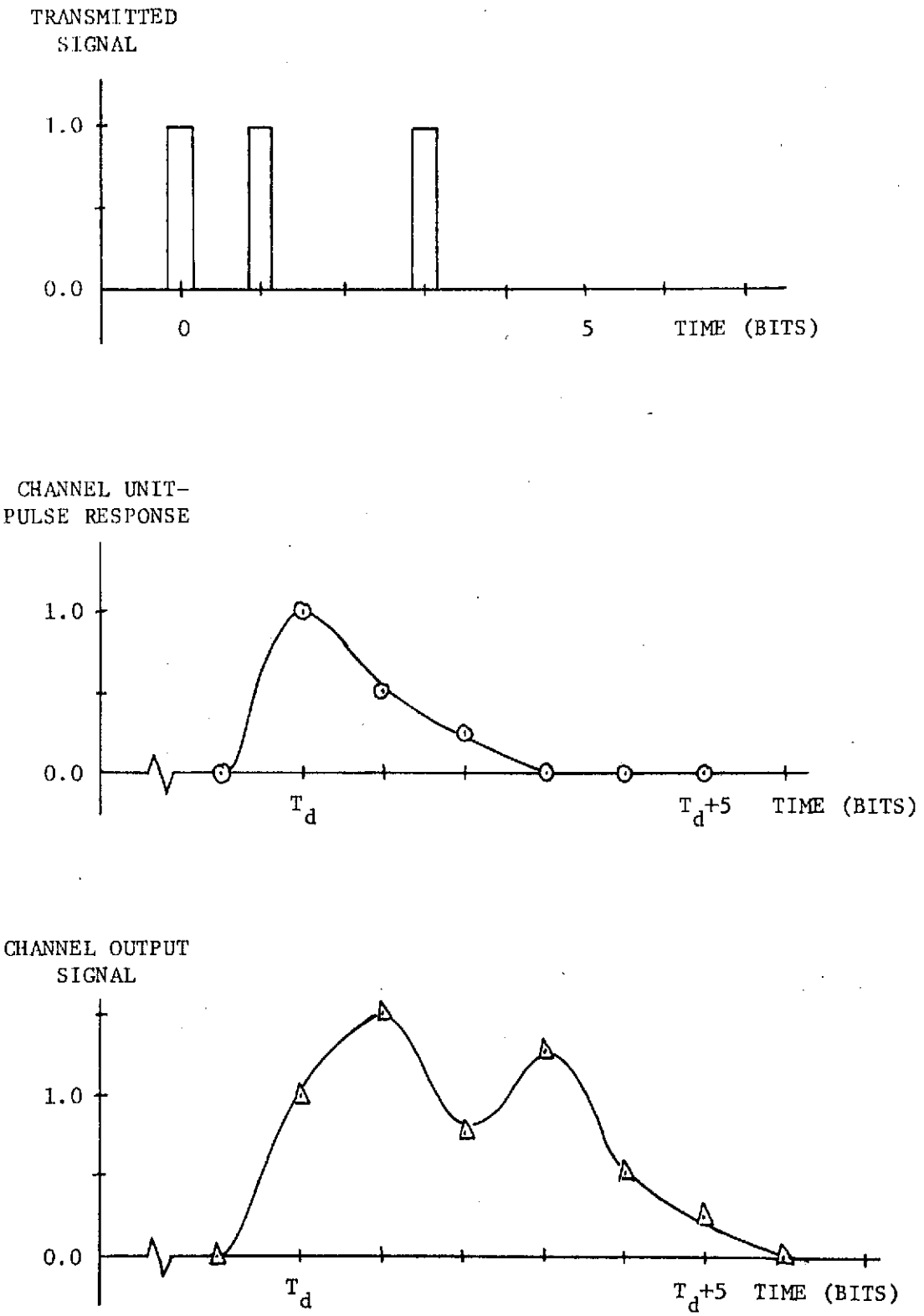

Fig. 1.1. Intersymbol Interference Due to Time Dispersion

A number of classification schemes have been proposed for equalizers; the most important for this study being analog or digital according to the type of signal being processed. Digital equalizers may be further classified by the duration of their response to a standard test signal called the unit pulse, i.e., a binary one preceded and followed by strings of binary zeros. The unit pulse is the discrete analog of the unit impulse for continuous systems, and is in fact often referred to as the "discrete unit impulse." If the theoretical unit-pulse response goes to zero after a finite time and remains there, then the equalizer has a finite impulse response (FIR), otherwise it has an infinite impulse response (IIR). Other classification schemes are based on linearity (linear, nonlinear), feedback (recursive, nonrecursive), signal flow (parallel, series, cascade), stationarity of characteristics (time-varying, time-invariant), and historical nomenclature (frequency-sampling, transversal, sampled data). A thorough discussion of these classifications is given by Rabiner, et al. [2]. Of particular interest here is the transversal filter structure, shown in Fig. 1.2. The blocks represent delay elements of duration T s, such as digital shift registers or analog tapped delay line stages, and the numbers $\{c_n\}$ are called tap-gains. It will be convenient to mathematically represent the T s delay by the z-transform delay operator z^{-1} which is discussed in A.4.† Transversal

†The notation A.4 refers to Section 4 of App. A.

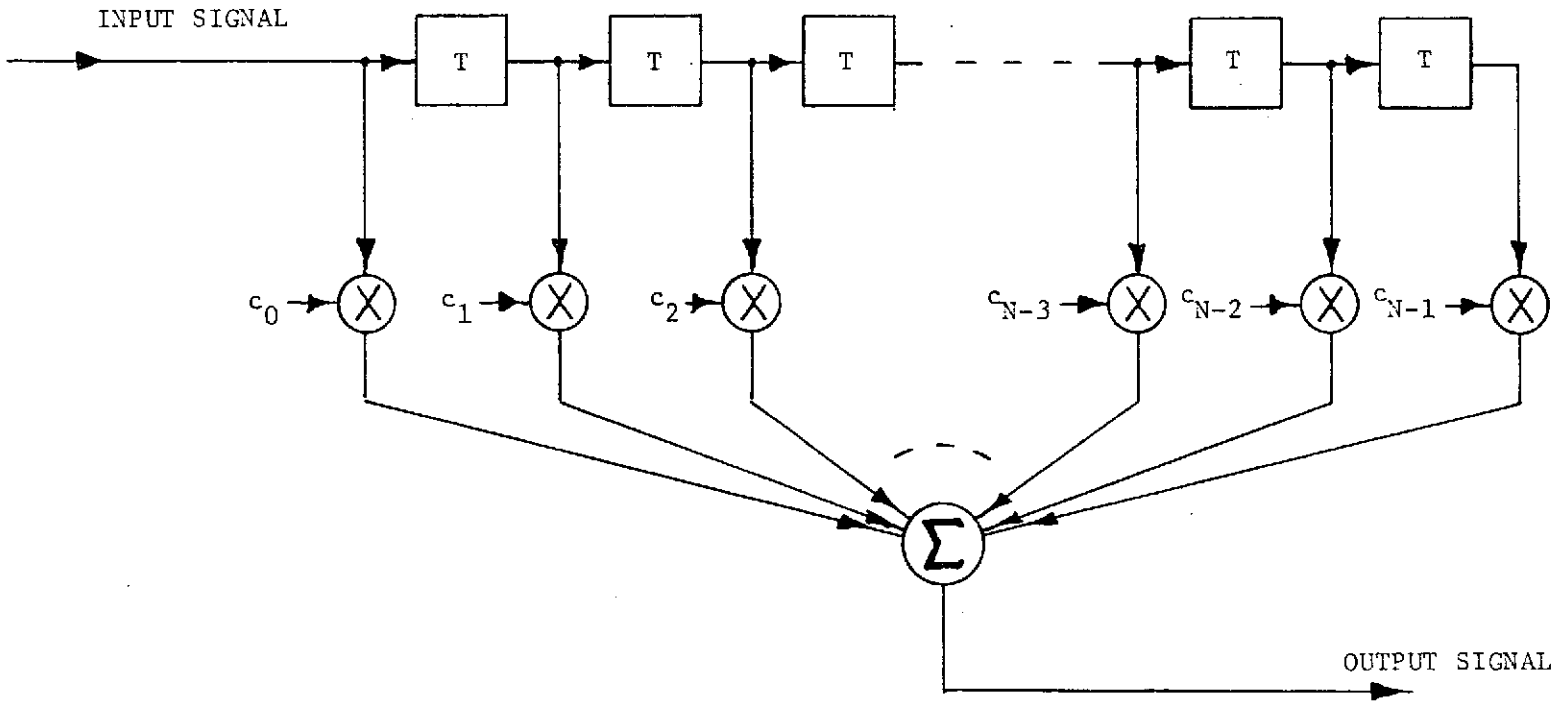

Fig. 1.2. The Transversal Filter Structure

equalizers have a number of nice properties such as good stability characteristics, ease of implementation, linearity, and mathematical tractability. Consequently, they have proven particularly useful in equalizing ISI.

If an equalizer is set at some initial time, e.g., when installed, and not changed thereafter, then it is called a preset or fixed equalizer. Fixed equalizers often prove inadequate in modern digital communications networks having switching between many branches, atmospheric transmission links, wide temperature fluctuations, and component aging. Such applications require an equalizer capable of being adjusted either during normal message transmission or at selected breaks between messages. Although the term adaptive equalization has sometimes been used for both of these cases, it will be reserved herein to describe only the former, while the latter case will be called automatic equalization [3]. Both adaptive and automatic equalizers will be referred to as self-adjusting.

Since a separate feedback channel from receiver to transmitter is usually infeasible in complex communication systems, some alternative method is necessary to produce an error signal for use by the equalizer. In adaptive operation this can be done by employing a decision-directed scheme. It is assumed that a high percentage of the output decisions are correct so that the sequence of output decisions closely approximates the transmitted sequence.

The error signal is then defined as the difference between signals before and after the decisions are made. The limitation of this scheme is obvious; if the equalizer is poorly adjusted initially, then the output sequence contains so many errors that the error signal becomes meaningless. Consequently, it is usually necessary to transmit an initial known training or test sequence which can be reproduced at the receiver. Comparison with the equalizer output before the decision device then gives the desired error signal. This known test sequence, transmitted during the initialization or training period is frequently selected to be a string of unit pulses with sufficient spacing between pulses to prevent the received waveforms from overlapping. After each such isolated test pulse an incremental adjustment of a particular set of equalizer parameters is made according to an iterative procedure called an algorithm. Assuming that the algorithm is convergent, the training period is continued until initialization is achieved, at which time the message transmission commences. In applications such as networks of sensors or polling systems where the messages are short, this training period can be considerably longer than the message length. Furthermore, many common equalizer structures require that the adjustment increments be small to insure convergence of the algorithm, thereby necessitating lengthy test sequences.

The problem to be studied herein is the development of an equalizer with a short initialization period that can be operated in an adaptive manner after initialization. It should have practical application to noisy time-varying networks that include switching. It is assumed throughout that sufficient digital computing equipment is available, such as a hardwired logic network, a minicomputer, or a time-shared general purpose computing system.

1.2 BACKGROUND AND ORGANIZATION

Some familiarity with the theory of modern digital communications in both the time and frequency domains is assumed herein, and a basic knowledge of applied mathematics is essential. The discrete nature of digital signals makes the z-transform a valuable tool, and some essential properties are given in App. A. This appendix also contains some fundamentals of linear operators which are then applied to the treatment of selected topics from the theory of vectors and matrices. Familiarity with probability and statistics is also desirable. A number of references are cited in the appendices both for completeness and as a source of proofs omitted from App. B. Finally, a reading knowledge of the FORTRAN programming language will be helpful in understanding the computer simulation listings found in App. C.

In addition to some introductory material on digital equalizers, this first chapter also contains a review of related work found in the literature. In Chapter 2 the initialization theory is developed for an automatic digital equalizer under the assumption that there is no background noise, i.e., the channel is deterministic. The equalizer uses a first-order gradient algorithm and a short test sequence to achieve initialization on the first iteration of the algorithm. A number of proofs relating to this chapter were relegated to App. B in order to preserve continuity. Chapter 3 first considers the effect of noise on the previously developed initialization theory. Then an approximation to the gradient algorithm is introduced which allows the equalizer to make on-line adjustments for time-variation of the channel parameters and noise statistics. The resulting adaptive equalizer is evaluated by means of a digital computer simulation, with the modeling assumptions presented and discussed. A complete description of the computer simulation program including a listing is found in App. C. The concluding chapter, Chapter 4, includes some practical considerations for implementation of the equalizer design developed in the preceding chapters and itemizes some additional areas for further investigation.

1.3 SURVEY OF RELATED LITERATURE

Equalization has been used on analog communication channels for many years to improve the channel amplitude vs. frequency characteristics. Such equalizers were generally of the fixed type, and the resulting channel phase vs. frequency characteristics were usually unimportant since the human ear is not very sensitive to the resulting delay distortion. However, with the advent in the 1950's of extensive digital traffic over the increasingly complex existing networks, it became more and more important to reduce ISI by making the phase vs. frequency characteristic as linear as possible in the channel passband. These attempts at increasing the data rates by decreasing ISI stimulated considerable interest in the use of transversal equalizers employing tapped delay lines and analog multipliers, e.g., attenuators, for the tap-gain multipliers. The first noteworthy attempt to make such equalizers self-adjusting was by Lucky [4] in 1965. By defining the peak distortion error criterion as a measure of ISI he obtained conditions for the optimal tap-gain settings using a series of isolated initializing pulses prior to message transmission in conjunction with a steepest-descent gradient technique. By computing the approximate gradient with only the polarity of the variables involved he was able to realize the equalizer using only digital logic. Lucky subsequently

combined the initialization procedure with a decision-directed on-line adaptive scheme for slowly-varying channels, and was again able to realize the equalizer with digital circuits [3]. Although these equalizers performed well when initialized, the initialization procedure generally required a large number of initializing pulses and did not converge at all when the peak distortion was greater than unity. Such equalizers were simple to implement and were inherently stable. They were named zero-forcing, since the unit-pulse response of the equalized channel was made zero at a number of points adjacent to the main pulse, with the number of such zeros dependent on the equalizer length.

In 1966 Widrow [5] introduced a transversal equalizer based on optimization with the mean-squared error criterion [6], and made it adaptive by using a decision-directed scheme and an appropriate "noisy" gradient estimate to update the tap-gains. Although this equalizer required linear multipliers rather than the less complex logic for computing the gradient, the speed of convergence was much faster. Initialization still required a large number of isolated initializing pulses. The application of this equalizer was first considered by Lucky and Rudin [7], and a detailed mathematical analysis of its convergence properties was done by Gersho [8]. An interesting all-digital serial realization of this type of equalizer was

developed by Lender [9]. Hirsch and Wolf [10] described an actual equalizer developed for Bell Labs using integrated circuits and field-effect transistors for the tap-gain multipliers. General treatments of these linear self-adjusting equalizers have been done [11,12], and the effects of quantization noise in digital implementations has also been studied [13].

In addition to the linear transversal equalizers, a number of other approaches have been made to the design and improvement of digital equalizers. One method [14] used the recursive frequency-sampling structure with a special initializing sequence that allowed each two-pole filter to be adjusted independently. Use of the Viterbi algorithm [15] for on-line channel identification for equalization was discussed by Magee and Prokias [16], and the somewhat related technique of dynamic programming has been suggested [17] to improve the initialization speed of an equalizer using a first-order gradient algorithm. By using the technique of quasilinearization the Kalman filter theory has been applied to estimate both the channel and the channel input [18]. Another approach to digital equalization uses the feedback of previous decisions through a transversal filter to compensate for the postcursors of the current pulse. This method is called decision-feedback [19,20], and is particularly good where the channel distortion is large with

relatively few precursors. However, decision errors tend to propagate and cause error bursts.

Throughout the development of digital equalizers there have been continuing attempts to reduce the time required for initialization. For short messages the initialization times were often excessive, e.g., an equalizer using the mean-squared error criteria was reported [10] to require several thousand isolated initializing pulses, and a considerably larger number of pulses was required when the equalization was done using the more slowly convergent zero-forcing equalizer. A number of methods were proposed to speed up this convergence, including the use of second-order gradient algorithms [21], use of different error criteria [22], and the periodic adjustment of the convergence factor in first-order algorithms [23,24]. This latter approach included the so-called "gearshift" method which used large initial adjustments to approach the optimal tap-gain value, and then changed to small adjustment increments for final accuracy. In 1971 Chang [25] proposed a new equalizer structure based on a set of parallel transversal filters. He showed that when the channel amplitude vs. frequency characteristic was accurately known, initialization was possible in one iteration of the tap-gain adjustment algorithm, and also that fast initialization was possible for a common class of partial response systems when the

channel was estimated. A similar structure has also been used to study the case when there are a small number of possible channels, each with a priori known statistics, to be equalized with a single equalizer [26]. There is currently a continuing interest in equalizer design, and the steady improvements in speed and availability of solid-state digital devices indicates that more sophisticated designs will soon be practical.

1.4 COMMENTS ON NOTATION

The theory of digital filters and equalizers has developed from a number of mathematical and engineering disciplines, e.g., networks, communications, control systems, optimization, and mathematical programming. Consequently, the notation and terminology found in the literature vary considerably. The most significant attempt at terminology standardization to date is found in Terminology in Digital Signal Processing [2] compiled by the committee on Digital Signal Processing of the IEEE Group on Audio and Electroacoustics.

A list of symbols used herein is found on page iv; however, some additional comments will also prove useful. All vectors are taken to be column vectors, and vector and matrix variables are indicated by an underline. To avoid confusion brackets are used to index the algorithm iterations. Several types of sequence notation are also

used. The sequences in Chapter 2 generally relate to the first iteration of the tap-gain adjustment algorithm and are indexed by subscripts, e.g. $\{x_k\}$, while those in Chapter 3 relate one-to-one with the algorithm iterations so the index is placed in brackets, e.g. {x[k]}. Sequences are also represented by the corresponding z-transform, e.g. X(z), in the manner discussed in A.4. When integrating in the z-domain the unit circle is always taken as the contour of integration. When evaluating these contour integrals it is often useful to make the substitution $z = e^{j\omega}$, where ω is the normalized frequency in radians per sample, i.e., $0 \leq \omega \leq 2\pi$. The resulting function will be denoted in the conventional manner, e.g., $X(j\omega)$ vice $X(e^{j\omega})$. Other notation that is specific to a particular section is defined therein and is not discussed here. This especially true of the appendices.

CHAPTER 2

A THEORY FOR ONE-STEP EQUALIZATION IN THE ABSENCE OF NOISE

After defining the channel model and the equalizer structure, an algorithm for equalizer adjustment is selected. The characteristics that control convergence of the transversal equalizer are studied, and from them a transformation is developed which insures convergence on the first iteration of the algorithm provided that the channel is noise free.

Before proceeding, however, it is necessary to briefly examine the overall digital communication system shown in Fig. 2.1. The transmitter produces a sequence of digital symbols $\{d_m\}$, taken here to be binary. This sequence is sent through a digital channel which distorts the input to produce the channel output sequence $\{x_m\}$. The digital channel is described by its discrete unit-pulse response $\{g_m\}$, but will usually include an analog channel such as a cable or radio link. It is then necessary to use a modulator at the analog channel input to map the digital sequence into a train of short analog waveforms, and a demodulator at the channel output to perform the

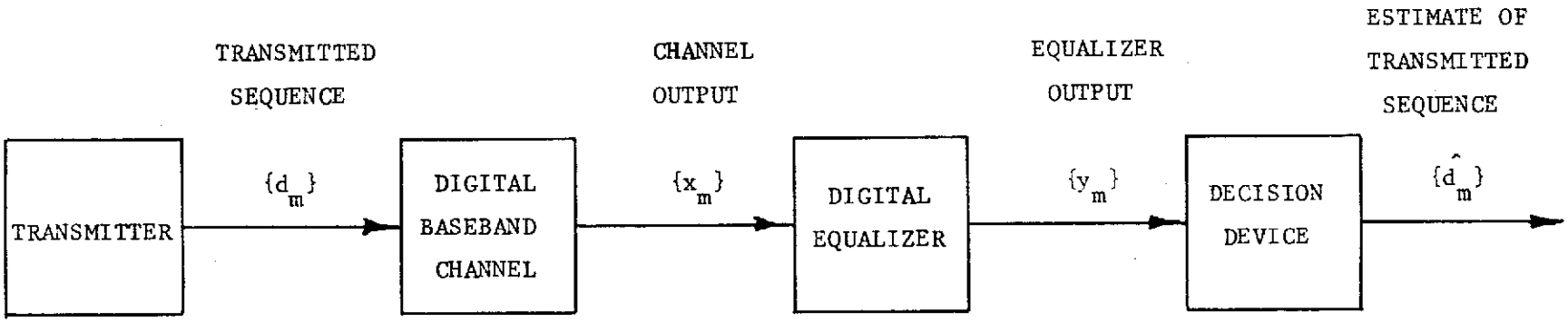

Fig. 2.1 Digital Communication System

inverse operation. In order to compensate for the effect of the channel an equalizer is employed at the channel output, and finally a decision device is used to obtain an estimate of the transmitted sequence based on observation of the equalizer output sequence $\{y_m\}$. Note that each block of Fig. 2.1 involves only digital inputs and outputs, hence these signals can be represented in the time domain by sequences, e.g., $\{x_m\}$, and in the transform domain by the respective z-transform

$$X(z) = \sum_{m=0}^{\infty} x_m z^{-m} \quad .$$

It will be assumed that all necessary timing and synchronization information is available, and that the channel time delay T_d is known. These problems are extensively covered in the literature [27,28]. The time delay will be absorbed in the notation; for example, the received signal x_k will correspond to the transmitted signal d_k although it actually occurs T_d s later.

2.1 CHANNEL MODEL AND THE GENERALIZED EQUALIZER STRUCTURE

The type of channel to be considered is a linear digital channel at baseband having a finite-length response $\{g_0, g_1, \ldots g_{G-1}\}$ to the unit pulse, where G is the sequence length. The channel will in general be time

varying due to switching and other background noise. For this chapter it will be assumed that the channel is free of stochastic noise, and that no switching occurs during the initialization period. Furthermore, other channel variations will be assumed slow in relation to the response time for a single pulse. These assumptions allow channel time variations to be neglected during the one-step initialization period. The effect of stochastic noise will be considered in Chapter 3. In addition to the unit-pulse response, the channel can be described by a transfer function

$$G(z) \triangleq \sum_{m=0}^{G-1} g_m z^{-m}, \qquad (2.1)$$

where $G(z)$ is the z-transform of $\{g_m\}$.

A fixed equalizer can in theory be located at any point of a linear channel. However, a self-adjusting equalizer is usually located at the channel output since it must obtain sufficient information about the channel to initialize itself and then maintain a suitable setting as the channel and noise characteristics change. When this channel identification is performed using a known test sequence or by a decision-directed scheme, the result is obtained at the channel output, and so any other equalizer placement necessitates a highly reliable feedback channel.

Many linear automatic equalizer designs can be put in the parallel form shown in Fig. 2.2, which is referred to as the generalized equalizer structure [7]. The blocks $\{F_m(z)\}$ will be called the filter-sections since each transfer function $F_m(z)$ represents a digital filter. When the channel output $X(z)$ is applied to the filter-sections, the equalizer output $Y(z)$ is obtained by multiplying each filter-section output $\{V_m(z)\}$ by the corresponding real-valued tap-gain c_m and then summing over the M weighted outputs $\{c_m V_m(z)\}$. The equalizer transfer function $H(z) \triangleq Y(z)/X(z)$ is determined as follows:

$$H(z) = \frac{1}{X(z)} \sum_{m=0}^{M-1} c_m V_m(z) = \underline{c}^T \underline{F}(z),$$

where $\underline{c} \triangleq (c_0, c_1, \ldots c_{M-1})^T$ and $\underline{F}(z) \triangleq (F_0(z), F_1(z), \ldots F_{M-1}(z))^T$. For self-adjusting operation the tap-gains are made adjustable. A device called an adapter seeks to minimize the ISI by iteratively computing a set of varying tap-gain increments $\{\Delta c_m[k]\}$ using a predetermined algorithm. The tap-gains are then adjusted at iteration k according to

$$c_m[k+1] = c_m[k] + \Delta c_m[k], \qquad m=0,1,\ldots M-1.$$

2.2 THE ALGORITHM AND ERROR CRITERION

The usefulness of the first-order gradient algorithm for self-adjusting equalizers is well established in the

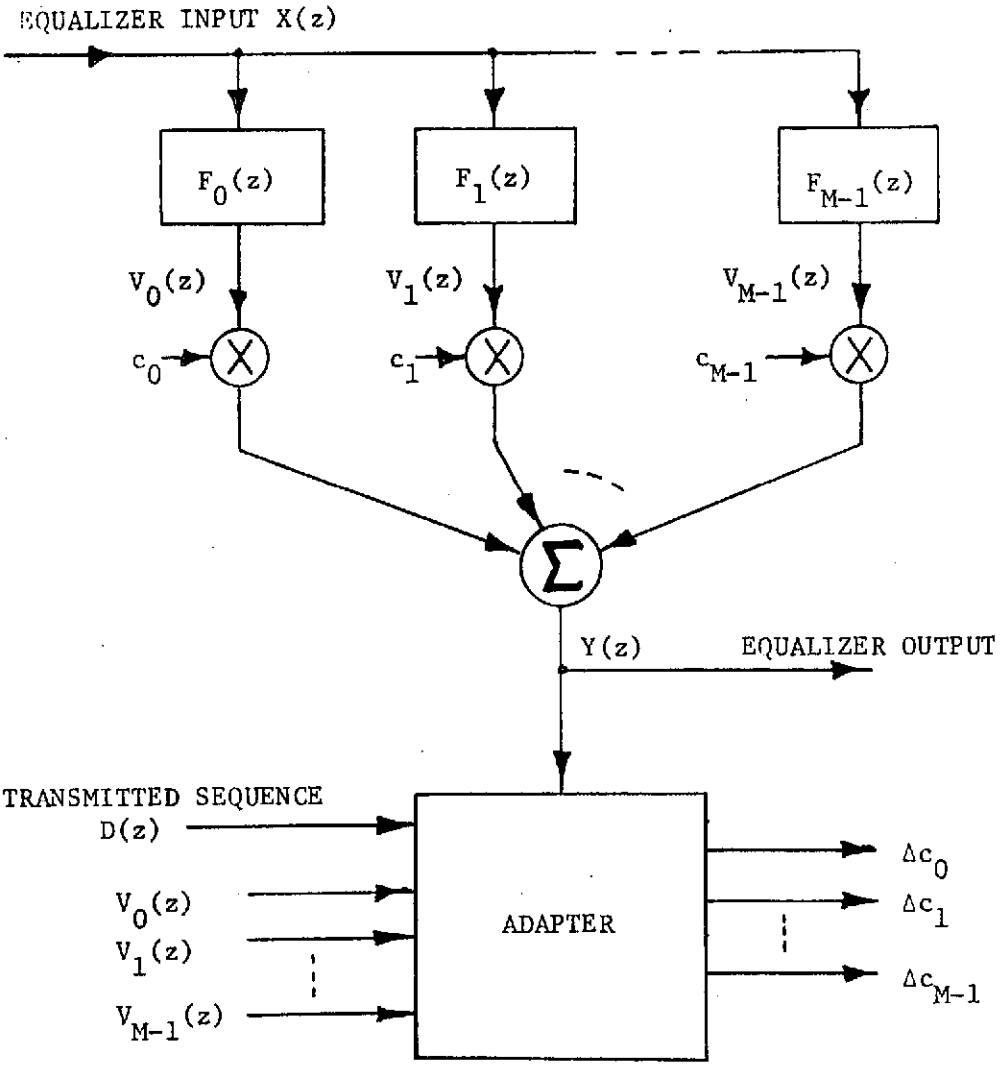

Fig. 2.2. The Generalized Equalizer Structure

literature [5,7,8] . It has the vector form

$$\underline{c}[k+1] = \underline{c}[k] - 1/2\ \gamma\ \nabla J(\underline{c}[k]), \tag{2.2}$$

where $\underline{c}[k]$ is the M-vector of tap-gains at iteration k and the corresponding tap-gain increment vector is given by $\Delta\underline{c}[k] = \underline{c}[k+1] - \underline{c}[k]$. The real scalar γ is called the convergence factor. It scales the tap-gain increments and thereby determines whether the sequence of tap-gain vectors $\{\underline{c}[k]\}$ converges to some final value, and if so, the rate of this convergence. The differentiable, real-valued function $J(\underline{c}[k])$, called the error criterion, is a quantitative measure of the equalizer performance in reducing ISI. Such a function is also called an index of performance, cost function, and objective function. The ISI remains constant when the gradient $\nabla J(\underline{c}[k]) = \underline{0}$, and from (2.2), no change is made in the tap-gains, i.e., the algorithm converges. Since the purpose of the algorithm is to determine a tap-gain vector $\hat{\underline{c}}$ which minimizes the ISI, it is necessary to define $J(\underline{c})$ so that the algorithm converges to $\hat{\underline{c}}$. One way of doing this is to select $J(\underline{c})$ to be convex. It then follows that any value $\hat{\underline{c}}$ satisfying $\nabla J(\hat{\underline{c}}) = \underline{0}$ also minimizes $J(\underline{c})$ [29].

Subject to these considerations, a convenient choice for the error criterion, analogous to the continuous integral-squared error [30], is the quadratic mean-squared error

(MSE) criterion given by

$$J(\underline{c}) = \sum_{m=-\infty}^{\infty} \varepsilon_m^2 \ , \qquad (2.3)$$

where $\varepsilon_m \triangleq y_m - d_m$ is the error corresponding to the transmitted symbol d_m. Using the complex convolution theorem (A.10), integrating around the unit circle of the z-plane, and noting that $\mathcal{E}(z) = Y(z) - D(z)$, it follows that

$$\begin{aligned} J(\underline{c}) &= \frac{1}{2\pi j} \oint \mathcal{E}(z^{-1})\, \mathcal{E}(z)\, z^{-1}\, dz \\ &= \underline{c}^T \underline{A}\, \underline{c} \ - 2\, \underline{c}^T \underline{b} + d, \end{aligned} \qquad (2.4)$$

where

$$\underline{A} \triangleq \frac{1}{2\pi j} \oint \underline{V}(z)\underline{V}^T(z^{-1}) z^{-1} dz = \frac{1}{2\pi j} \oint X(z)\underline{F}(z)\underline{F}^T(z^{-1})X(z^{-1}) z^{-1} dz$$

$$\underline{b} \triangleq \frac{1}{2\pi j} \oint \left[\underline{F}(z)\ X(z)\ D(z^{-1}) + \underline{F}(z^{-1})X(z^{-1})D(z)\right]\ z^{-1} dz \qquad (2.5)$$

and

$$d \triangleq \frac{1}{2\pi j} \oint D(z)\ D(z^{-1}) z^{-1} dz.$$

By letting $z = e^{j\omega}$ it is easy to show that both the M-vector $\underline{b}$ and the scaler d are real. The MxM matrix $\underline{A}$ in (2.5) is shown to be hermitian and positive semidefinite in B.1. To insure that $\underline{A}$ is also real, hence real symmetric, it is assumed that the filter-sections are FIR filters so that the output of $F_m(z)$ corresponding to input X(z) has the form

$$V_m(z) = \sum_{k=0}^{K-1} v_{mk} z^{-k}. \tag{2.6}$$

This assumption will be shown valid in (2.26) for the particular equalizer considered. Now from (1.5) it follows that any arbitrary element a_{mn} of $\underline{A}$ is real since

$$a_{mn} = \frac{1}{2\pi j} \oint V_m(z)\ V_n(z^{-1}) z^{-1} dz = \sum_{k=0}^{K-1} v_{mk}\ v_{nk}. \tag{2.7}$$

Since $\underline{A}$ is positive semidefinite, $J(\underline{c})$ is a convex function. Thus the gradient algorithm is applicable and the gradient is given by

$$\nabla J(\underline{c}) = 2\ \underline{A}\ \underline{c} - 2\ \underline{b}. \tag{2.8}$$

The unique minimum value of ISI thus occurs for any tap-gain vector $\hat{\underline{c}}$ satisfying $\nabla J(\hat{\underline{c}}) = \underline{0}$, or equivalently when

$$\underline{A}\ \hat{\underline{c}} = \underline{b}\ . \tag{2.9}$$

2.3 THE TRANSVERSAL EQUALIZER

Since $\underline{A}$ is positive semidefinite it may be singular, in which case there will not be a unique solution $\hat{\underline{c}}$ for (2.9). However, it is possible to select the filter-sections so that $\underline{A}$ is strictly positive definite, as will be shown. In particular, if $F_m(z) = z^{-m}$ for m=0,1,...M-1, and all filter-sections share the same shift register, then a

transversal equalizer results. To distinguish this case the quantities $\underline{F}(z)$, $\underline{V}(z)$, $\underline{A}$, and $\underline{b}$ in (2.5) will be identified by a tilde, e.g., $\tilde{\underline{A}}$. It is shown in B.2 that $\tilde{\underline{A}}$ is positive definite, implying that it is nonsingular. Thus $\tilde{\underline{A}}^{-1}$ exists and the optimal tap-gain vector is found to be $\hat{\underline{c}} = \tilde{\underline{A}}^{-1} \tilde{\underline{b}}$. The tap-gain error vector $\underline{e}$ is now defined by

$$\underline{e} \triangleq \underline{c} - \hat{\underline{c}} = \underline{c} - \tilde{\underline{A}}^{-1} \tilde{\underline{b}}, \tag{2.10}$$

and is given at iteration k by $\underline{e}[k] = \underline{c}[k] - \hat{\underline{c}}$. By adding and subtracting the term $\tilde{\underline{b}}^T \tilde{\underline{A}}^{-1} \tilde{\underline{b}}$, (2.4) can be reformulated as

$$J(\underline{c}[k]) = d - \tilde{\underline{b}}^T \tilde{\underline{A}}^{-1} \tilde{\underline{b}} + \underline{e}^T[k] \tilde{\underline{A}} \underline{e}[k]. \tag{2.11}$$

The term $\underline{e}^T[k] \tilde{\underline{A}} \underline{e}[k]$ is a positive definite quadratic form, which is zero only when $\underline{e}[k] = \underline{0}$. In this case (2.10) implies that $\underline{c}[k] = \hat{\underline{c}}$, so the <u>irreducable error</u> J_{min} equals

$$J_{min} = d - \tilde{\underline{b}}^T \tilde{\underline{A}}^{-1} \tilde{\underline{b}}. \tag{2.12}$$

The correctable error after k iterations is

$$J_{cor}[k] = \underline{e}^T[k] \tilde{\underline{A}} \underline{e}[k]. \tag{2.13}$$

Substituting (2.8) into (2.2) and using (2.10) gives the algorithm as

$$\underline{c}[k+1] = \underline{c}[k] - \gamma \tilde{\underline{A}} \underline{e}[k].$$

Repeated substitution into (2.10) with decreasing index k results in the "telescoping" form

$$\underline{e}[k] = \left[\prod_{i=0}^{k-1} (\underline{I} - \gamma \tilde{\underline{A}})\right] \underline{e}[0]$$
$$= (\underline{I} - \gamma \tilde{\underline{A}})^k \underline{e}[0]. \qquad (2.14)$$

Further simplification results from the observation that, since $\tilde{\underline{A}}$ is real symmetric, all eigenvalues are real and there exists a unitary transformation $\tilde{\underline{Q}}$ such that $\tilde{\underline{Q}}^T \tilde{\underline{A}} \tilde{\underline{Q}} = \tilde{\underline{\Lambda}}$, where the diagonal elements of $\tilde{\underline{\Lambda}} = \text{diag}(\tilde{\lambda}_0, \tilde{\lambda}_1, \ldots \tilde{\lambda}_{M-1})$ are the eigenvalues of $\tilde{\underline{A}}$ and the columns of $\tilde{\underline{Q}}$ are the corresponding normalized eigenvectors $\{\tilde{\underline{q}}_m\}$ of $\tilde{\underline{A}}$. Upon making the substitution $\tilde{\underline{A}} = \tilde{\underline{Q}} \tilde{\underline{\Lambda}} \tilde{\underline{Q}}^T$ (2.14) becomes

$$\underline{e}[k] = \tilde{\underline{Q}} (\underline{I} - \gamma \tilde{\underline{\Lambda}})^k \tilde{\underline{Q}}^T \underline{e}[0],$$

which when substituted into (2.13) gives

$$J_{cor}[k] = \sum_{m=0}^{M-1} (\underline{e}^T[0]\tilde{\underline{q}}_m)^2 \tilde{\lambda}_m (1-\gamma\tilde{\lambda}_m)^{2k} . \qquad (2.15)$$

It has been observed by Chang [25] that the eigenvalue spread of $\tilde{\underline{A}}$, i.e., the difference between the largest and smallest eigenvalues, is a crucial factor in the rate of convergence. If all eigenvalues are equal then J_{cor} can be reduced to zero on the first iteration by setting $\gamma = \tilde{\lambda}_m^{-1}$. Unfortunately, since $\tilde{\underline{A}}$ depends on the input sequence to the equalizer, there is no general way to control the eigenvalues without changing the

equalizer structure. A logical approach is to investigate a more suitable form for the filter-sections $\{F_m(z)\}$, and this will be done in the next section. It is interesting to note that by selecting a more general algorithm the correctable error can be reduced to zero on the first iteration using the transversal structure. This case is detailed in B.3.

2.4 THE GENERALIZED TRANSVERSAL EQUALIZER

Perhaps the simplest extention of the transversal equalizer is that generalized equalizer structure in which each filter-section is itself a transversal filter. Such an equalizer will be called a generalized transversal equalizer, and a typical filter-section is shown in Fig. 2.3. This equalizer can be implemented so that all filter-sections share a common shift register. For convenience it will be assumed that all filter-sections are of equal length N, which will be strictly less than M. The resulting filter-sections are described by

$$F_m(z) = \sum_{n=0}^{N-1} f_{mn} z^{-n} \quad , \; m = 0, 1, \ldots M-1. \quad (2.16)$$

They can be considered as devices which form linear combinations of those N values of the channel output sequence contained in the equalizer shift register, or equivalently, as the following linear time-invariant transformation on this sequence.

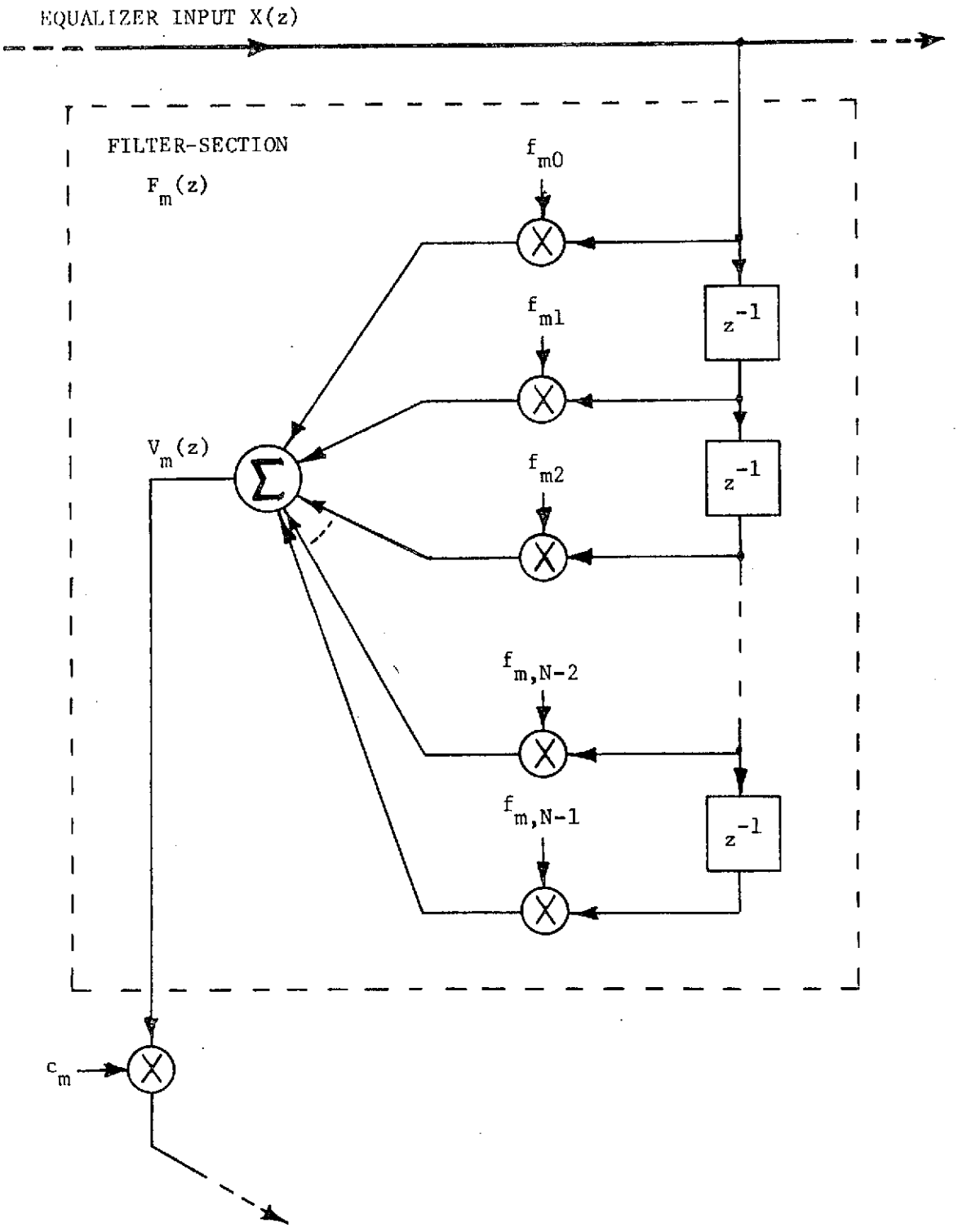

Fig. 2.3. Generalized Transversal Equalizer Filter-Section $F_m(z)$

$$\underline{F}(z) = \sum_{n=0}^{N-1} \begin{bmatrix} f_{0n} \\ f_{1n} \\ \cdot \\ \cdot \\ \cdot \\ f_{M-1,0} \end{bmatrix} z^{-n} = \begin{bmatrix} f_{00} & f_{01} & \cdots & f_{0,N-1} \\ f_{10} & f_{11} & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \\ f_{M-1,0} & & \cdots & f_{M-1,N-1} \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ \cdot \\ \cdot \\ \cdot \\ z^{-N+1} \end{bmatrix} \tag{2.17}$$

The MxN matrix in (2.17) is now augmented by adding M-N zeros to each row, and the resulting MxM matrix is denoted $\underline{F}^T$. Notice that column m of $\underline{F}$ corresponds to the coefficients of $F_m(z)$. This corresponds to physically extending the shift-register length from N to M and assigning tap-gains of zero to the new tap-gain multipliers, and is done strictly for mathematical convenience. Upon recalling that for the transversal case

$\tilde{\underline{F}}(z) = (1, z^{-1}, z^{-2}, \ldots z^{-M+1})^T$, the relation

$$\underline{F}(z) = \underline{F}^T \tilde{\underline{F}}(z) \tag{2.18}$$

is seen to hold. Noting that $\underline{V}(z) = \underline{F}^T \tilde{\underline{F}}(z)\ X(z)$ $= \underline{F}^T \tilde{\underline{V}}(z)$ it follows from (2.5) that $\underline{A} = \underline{F}^T \tilde{\underline{A}}\ \underline{F}$ and $\underline{b} = \underline{F}^T \tilde{\underline{b}}$. Substituting these expressions into (2.4) gives the new MSE expression

$$J(\underline{c}) = \underline{c}^T \underline{F}^T \tilde{\underline{A}}\ \underline{F}\ \underline{c} - 2\ \underline{c}^T \underline{F}^T \tilde{\underline{b}} + d. \tag{2.19}$$

Taking the gradient of (2.19) gives

$$\nabla J(\underline{c}) = 2\ \underline{F}^T\ \tilde{\underline{A}}\ \underline{F}\ \underline{c} - 2\ \underline{F}^T\ \tilde{\underline{b}} \triangleq 2\ (\underline{A}\ \underline{c} - \underline{b})\ , \qquad (2.20)$$

Thus any vector $\hat{\underline{c}}$ satisfying $\underline{A}\ \hat{\underline{c}} = \underline{b}$ minimizes the ISI and is consequently an optimal tap-gain vector. It is crucial to note here that such a vector exists even though $\underline{A}$ is singular because, as proved in B.4, $\underline{b}$ is always in the range space of $\underline{A}$. The optimal tap-gain vector in this case is given by $\hat{\underline{c}} = \underline{A}^{-I}\ \underline{b}$, where $\underline{A}^{-I}$ is the pseudoinverse of $\underline{A}$ as defined in A.3.

By adding and subtracting the term $\underline{b}^T\ \underline{A}^{-I}\ \underline{b}$ in (2.20) and using the facts from A.3 that $\underline{A}\ \underline{A}^{-I}\ \underline{b} = \underline{b}$ and $\underline{A}\ \underline{A}^{-I}\ \underline{A} = \underline{A}$, the error criterion can be reformulated in terms of the tap-gain error vector

$$\underline{e} = \underline{c} - \hat{\underline{c}} = \underline{c} - \underline{A}^{-I}\ \underline{b}. \qquad (2.21)$$

This gives

$$J(\underline{c}) = d - \underline{b}^T\ \underline{A}^{-I}\ \underline{b} + \underline{e}^T\ \underline{A}\ \underline{e}. \qquad (2.22)$$

The term $d - \underline{b}^T\ \underline{A}^{-I}\ \underline{b}$ is independent of the tap-gain vector $\underline{c}$ and represents the minimum error $J(\hat{\underline{c}}) \triangleq J_{min}$, while the quadratic term again represents the correctable error J_{cor}. The symmetry of $\underline{A}$ follows from that of $\tilde{\underline{A}}$ upon noting that, for arbitrary m and n, $\tilde{a}_{ik} = \tilde{a}_{ki}$ implies

$$a_{mn} = \sum_{i=0}^{M-1} \sum_{k=0}^{M-1} f_{im}\, \tilde{a}_{ik}\, f_{kn}$$

$$= \sum_{i=0}^{M-1} \sum_{k=0}^{M-1} f_{kn}\, \tilde{a}_{ki}\, f_{im} = a_{nm}.$$

Therefore, the matrix $\underline{Q}$ having the normalized eigenvectors $\{\underline{q}_m\}$ of $\underline{A}$ as columns forms a unitary transformation that diagonalizes $\underline{A}$ according to $\underline{Q}^T \underline{A}\, \underline{Q} = \underline{\Lambda} = \text{diag}(\lambda_0, \lambda_1, \ldots \lambda_{M-1})$ with $\{\lambda_m\}$ being the set of eigenvalues of $\underline{A}$. By substituting the gradient (2.20) into the algorithm and recognizing $\underline{e}[k]$ at iteration k from (2.21) it follows that

$$\underline{c}[k+1] = \underline{c}[k] - \gamma\, \underline{A}\, \underline{e}[k]. \tag{2.23}$$

Subtracting $\hat{\underline{c}}$ from both sides of (2.23) gives

$$\underline{e}[k+1] = \underline{e}[k] - \gamma\, \underline{A}\, \underline{e}[k],$$

which reduces to the following expression in terms of $\underline{e}[0]$,

$$\underline{e}[k+1] = \underline{Q}(\underline{I} - \gamma\, \underline{\Lambda})^k\, \underline{Q}^T \underline{e}[0].$$

Finally, $J_{cor}[k]$ expressed in terms of this result is

$$J_{cor}[k] = (\underline{Q}^T \underline{e}[0])^T (\underline{I} - \gamma\, \underline{\Lambda})^k\, \underline{\Lambda} (\underline{I} - \gamma\, \underline{\Lambda})^k\, (\underline{Q}^T\, \underline{e}[0])$$

$$= \sum_{m=0}^{M-1} (\underline{e}^T[0]\, \underline{q}_m)^2\, \lambda_m\, (1-\gamma\, \lambda_m)^{2k}. \tag{2.24}$$

Since $\underline{F}$, and hence $\underline{A}$, is singular then one or more of the eigenvalues λ_m will be zero and the corresponding terms of (2.24) are then zero, too. This observation will prove useful in the next section.

If an input sequence $\{d_m\}$ of length D is transmitted over the digital channel of (2.1) then the channel output is given in the transform domain by

$$X(z) = G(z)\ D(z) = \sum_{n=0}^{G-1} \sum_{m=0}^{D-1} g_n\ d_m\ z^{-(m+n)} \quad .$$

Rearranging the summations and letting k = m+n results in

$$X(z) = \sum_{k=0}^{D+G-2} \left[\sum_{n=0}^{k} g_n\ d_{k-n} \right] z^{-k} \qquad (2.25)$$

$$\triangleq \sum_{k=0}^{L-1} x_k\ z^{-k},$$

where $L \triangleq D + G - 1$ is the length of the channel output sequence. The channel output sequence $\{x_m\}$ is given by the discrete convolution of the channel input $\{d_m\}$ and channel unit-pulse response $\{g_m\}$, as seen from (2.25). An analogous derivation for each filter-section of length N gives the filter-section output

$$V_m(z) = \sum_{k=0}^{L+N-2} \left[\sum_{n=0}^{k} f_{mn}\ x_{k-n} \right] z^{-k} \triangleq \sum_{k=0}^{K-1} v_{mk} z^{-k}, \qquad (2.26)$$

where the sequence $\{v_{mk}\}$ has length $K \triangleq L + N - 1$.

At this point it is necessary to make some assumptions which will lead to the selection of the filter-section length N and the number of such sections M. First, a finite upper limit $\hat{L}$ of the channel output sequence length L is assumed known. Since a FIR channel was assumed in Section 2.1 and the transmitted initializing sequence $\{d_m\}$ is known, $\hat{L}$ certainly exists. Second, it is assumed that M is at least as large as the length of the filter-section output sequences K, i.e.,

$$M \geq N + \hat{L} - 1 \ . \tag{2.27}$$

For channels with $\hat{L}$ very large it may be necessary to truncate the channel unit-pulse response in order to obtain a reasonable value for M. This has only a slight effect on performance when the truncated terms are small. This last condition (2.27) insures that the output sequence length can always be extended from K to M by appending M-K zeros to the transmitted sequence $\{d_m\}$. The selection of numerical values for M and N will be considered in the next section.

The convolution relation (2.26) can now be written in the following matrix form

$$\begin{bmatrix} v_{m0} \\ v_{m1} \\ \vdots \\ v_{m,K-1} \\ \vdots \\ v_{m,M-1} \end{bmatrix} = \begin{bmatrix} x_0 & 0 & \cdots & 0 \\ x_1 & x_0 & & \vdots \\ \vdots & x_1 & & \\ x_{L-1} & \vdots & & \\ 0 & x_{L-1} & & x_0 \\ \vdots & & & x_1 \\ & 0 & & \vdots \\ & \vdots & & \\ & & & \end{bmatrix} \begin{bmatrix} f_{m0} \\ f_{m1} \\ \vdots \\ f_{m,N-1} \end{bmatrix}$$

or more concisely as

$$\underline{v}_m = \underline{X}\, \underline{f}_m \,, \quad m = 0, 1, \ldots M-1. \tag{2.28}$$

The MxN matrix $\underline{X}$ has rank N since the NxN principle minor is lower triangular with nonzero diagonal elements. The filter-section output vector $\underline{v}_m$ is an M-vector, and the filter-section tap-gain vector $\underline{f}_m$ has length N. With this vector notation an arbitrary element a_{mn} of $\underline{A}$ can be written using (2.7) as

$$a_{mn} = \langle \underline{v}_m, \underline{v}_n \rangle \,, \tag{2.29}$$

which is the correlation of the output sequences from filter-sections m and n, i.e., $\underline{A}$ is a correlation matrix. It can now be seen from (2.24) that J_{cor} is reduced to zero on the first iteration when all nonzero eigenvalues of $\underline{A}$ are equal, since the convergence factor need only be set at $\gamma = \lambda_m^{-1}$ for $\lambda_m \neq 0$. Conditions for which $\lambda_m = 0$ or $\lambda_m = \gamma^{-1}$ are established in the next section.

2.5 CONDITIONS FOR ONE-STEP INITIALIZATION

Since the elements of $\underline{A}$ depend on the filter-section output vectors $\{\underline{v}_m\}$, their relation to the eigenvalues of $\underline{A}$ is most important. An ideal solution would be to determine a set $\{\hat{\underline{f}}_m\}$ of *ideal* filter-section tap-gain vectors that would make the corresponding set $\{\hat{\underline{v}}_m\}$ orthonormal for a given channel output. It then follows from (2.29) that $\underline{A}$ is the identity matrix with all eigenvalues equal to unity. Unfortunately, (2.28) represents M scalar equations in N unknowns with N < M which cannot in general be solved exactly for $\underline{f}_m$. A "best" approximate solution does exist, however, and is given by

$$\underline{f}_m = \underline{X}^{-I}\, \hat{\underline{v}}_m \ , \tag{2.30}$$

where $\underline{X}^{-I}$ is the pseudoinverse of $\underline{X}$ as defined in A.3, and $\underline{f}_m$ is the *actual* filter-section tap-gain vector. The actual filter-section output vector $\underline{v}_m$ and the ideal vector $\hat{\underline{v}}_m$ which it approximates can be related by

$$\underline{v}_m = \underline{X}\, \underline{f}_m = \underline{X}\, \underline{X}^{-I}\, \hat{\underline{v}}_m \ . \tag{2.31}$$

From A.3 $(\underline{X}\, \underline{X}^{-I})^T\, \underline{X}\, \underline{X}^{-I} = \underline{X}\, \underline{X}^{-I}$, and substitution of this result along with (2.31) into (2.29) gives

$$a_{mn} = \left\langle \underline{X}\, \underline{X}^{-I}\, \hat{\underline{v}}_m,\ \underline{X}\, \underline{X}^{-I}\, \hat{\underline{v}}_n \right\rangle = \left\langle \hat{\underline{v}}_m,\ \underline{X}\, \underline{X}^{-I}\, \hat{\underline{v}}_n \right\rangle . \tag{2.32}$$

Using the orthonormal sets $\{\underline{\psi}_0, \underline{\psi}_1, \ldots \underline{\psi}_{M-1}\}$ and $\{\underline{\xi}_0, \underline{\xi}_1, \ldots \underline{\xi}_{N-1}\}$ and the generalized spectral representations for $\underline{X}$ and $\underline{X}^{-I}$ as defined in A.3, it is easy to show that

$$\underline{X}\ \underline{X}^{-I} = \sum_{m=0}^{N-1} \underline{\psi}_m\ \underline{\psi}_m^T \quad . \tag{2.33}$$

Furthermore, the orthonormal set $\{\underline{\psi}_m\}$ is linearly independent and thus constitutes a basis for the real vector space of dimension M to which $\underline{y}_m$ and $\hat{\underline{y}}_m$ belong. Consequently, $\hat{\underline{y}}_m$ can be represented in this basis by

$$\hat{\underline{y}}_m = \sum_{n=0}^{M-1} h_{mn}\ \underline{\psi}_n \ , \tag{2.34}$$

where $h_{mn} = \langle \hat{\underline{y}}_m, \underline{\psi}_n \rangle$ is the coordinate in the direction of $\underline{\psi}_n$. Now let $\underline{H}$, $\hat{\underline{Y}}$ and $\underline{\Psi}$ be the matrices with columns coinciding with the ordered sets $\{h_{mn}\}$, $\{\hat{\underline{y}}_m\}$, and $\{\underline{\psi}_m\}$, respectively. Then making $\underline{H}$ an orthogonal matrix also makes $\hat{\underline{Y}}$ orthogonal, as is easily shown by observing that $\underline{\Psi}$ is unitary or equivalently, that

$$\langle \hat{\underline{y}}_m, \hat{\underline{y}}_n \rangle = \sum_{k=0}^{M-1} \sum_{i=0}^{M-1} h_{mk}\ h_{ni} \langle \underline{\psi}_k, \underline{\psi}_i \rangle$$

$$= \sum_{k=0}^{M-1} h_{mk}\ h_{nk} = \langle \underline{h}_m, \underline{h}_n \rangle$$

for all m and n. Substituting (2.33) and (2.34) into (2.32) gives an expression for any element a_{mn} of $\underline{A}$ in terms of the set $\{\underline{h}_m\}$,

$$a_{mn} = \sum_{k=0}^{N-1} h_{mk}\, h_{nk} \ . \tag{2.35}$$

To make the set $\{\underline{h}_m\}$ orthonormal the matrix $\underline{H}$ must be orthogonal. Before discussing the selection of such a matrix, however, the eigenvalues of $\underline{A}$ will be determined.

By defining the MxM matrix $\underline{V}$ as having columns coinciding with $\{\underline{v}_m\}$ and letting $\underline{P} \triangleq \underline{X}\ \underline{X}^{-I}$, (2.31) and (2.34) can be combined into the matrix equation

$$\underline{V} = \underline{P}\ \hat{\underline{V}} = \underline{P}\ \underline{\Psi}\ \underline{H} \ .$$

Furthermore, observing from (2.29) that $\underline{A}$ can be written as $\underline{A} = \underline{V}^T\ \underline{V}$, it follows that

$$\underline{A} = \underline{H}^T\ \underline{\Psi}^T\ \underline{P}^T\ \underline{P}\ \underline{\Psi}\ \underline{H} \ . \tag{2.36}$$

It is shown in B.5 that $\underline{P}$ represents a projection operator so that $\underline{P}^T\ \underline{P} = \underline{P}^2 = \underline{P}$, and in B.6 that $\underline{\Psi}^T\ \underline{P}\ \underline{\Psi}$ equals the block diagonal matrix $\mathrm{diag}(\underline{I}_N \mid \underline{0}_{M-N})$. Since $\underline{H}$ is orthogonal, (2.36) implies that $\underline{A}$ is similar to $\mathrm{diag}(\underline{I}_N \mid \underline{0}_{M-N})$ and thus has the same eigenvalues, namely, N equal to unity and M-N equal to zero. The desired conditions for one-step

initialization have now been found, as can be seen from setting $\gamma = 1$ in (2.24) and observing that each term of the summation is then zero.

There are an infinite number of orthogonal matrices which could be used for $\underline{H}$, however the class of Hadamard matrices, discussed in A.2, is particularly well suited for generating $\underline{H}$. Such matrices are binary in the sense that all elements are either +1 or -1, have orthogonal columns, and are known to exist for all orders which are multiples of four over the range of feasible values of M. Consequently, normalizing an MxM Hadamard matrix by $1/\sqrt{M}$ produces an orthogonal matrix with all elements equal in magnitude. This is particularly convenient for digital computation since the normalization can often be done near the end of the computation rather than on each element of $\underline{A}$ initially.

The filter-section tap-gains can be computed without actually computing the pseudoinverse $\underline{X}^{-I}$, since substituting (2.33) and (2.34) into (2.30) gives

$$\underline{f}_m = \left(\sum_{k=0}^{N-1} \mu_k^{-1} \, \underline{\xi}_k \, \underline{\psi}_k^T \right) \left(\sum_{n=0}^{M-1} h_{mn} \, \underline{\psi}_n \right) \tag{2.37}$$

$$= \sum_{k=0}^{N-1} \mu_k^{-1} \, h_{mk} \, \underline{\xi}_k \, , \quad m = 0, 1, \ldots M-1.$$

This expression involves only the first N rows of $\underline{H}$ and the eigenvalues $\{\mu_m^2\}$ and eigenvectors $\{\underline{\xi}_m\}$ of the NxN real symmetric matrix $\underline{X}^T \underline{X}$ as discussed in A.3. A number of numerical methods exist for finding eigenvalues and eigenvectors which are particularly simple for real symmetric matrices. Some of these methods are discussed in C.3.

2.6 SIMULATION RESULTS FOR NOISE-FREE CHANNELS

The equalizer initialization theory developed in this chapter was tested by using a digital computer simulation. A detailed description of the simulation, including a complete listing, is found in App. C. Since this simulation was written for a maximum of 32 filter-sections, i.e., $M \leq 32$, there is a practical limit to the length of the channel unit-pulse response as indicated by (2.27). Under this restriction, three digital channel models were selected for their representative characteristics. In the absence of an actual data channel, two models were taken from the literature and one was determined in the laboratory.

The first channel was a low-pass filter (LPF) having a normalized cutoff frequency of 1.0 Hz and a 40 dB/decade roll-off. It is described mathematically by the Fourier transform pair

$$G_1(f) = (1+jf)^{-2} \leftrightarrow g_1(t) = 4\pi^2 \, te^{-2\pi t}, \; t \geq 0. \quad (2.38)$$

By sampling at 0.1 s intervals and truncating all sample values which are less than 0.02, the unit-pulse response {2.11, 2.25, 1.80, 1.28, 0.85, 0.55, 0.34, 0.21, 0.12, 0.07, 0.04, 0.02} of length 12 is obtained. Although this channel is rather artificial for actual data communications, it is useful to demonstrate equalizer performance with nonnegative samples which are relatively large with respect to unity. It also serves for comparison with an ideal bandlimited channel and the previous work of Burlage and Houts [31]. An experimentally convenient and widely used measure of ISI is the binary eye opening and the related peak-distortion criterion [32]. Assuming that the second sample corresponds to the desired unity output, then the normalized peak distortion is 3.284, and thus considerably more than the value of 0.5 required for eye-closure.

For the second channel model an electronic bandpass filter (BPF) [33] was used which had a maximally-flat passband with break frequencies of 200 Hz and 3.0 kHz and 60 dB/decade roll-off. The unit-pulse response of this channel was determined in the laboratory by exciting the filter with white noise and crosscorrelating the filter output with delayed versions of the white noise. Assuming a sampling interval of 50 μs, the unit-pulse of length 11

was {0.50, 1.15, 0.50, -0.25, -0.50, -0.35, -0.20, -0.10, -0.15, -0.10, -0.05}. This channel passband approximates the amplitude characteristic of a voice-grade telephone line, with zero phase at midband and tending to ± 270° in the stopbands. Since the normalized peak distortion is 2.348, the binary eye is again well closed.

The third channel model was taken from a paper by Wescott [34], and represents the unit-pulse response of a vestigial-sideband (VSB) amplitude-modulated data link composed of two carrier systems and 100 miles of loaded cable. When sampled at 0.4167 ms, corresponding to a transmission rate of 2400 baud, the unit-pulse response of length 9 was reported to be {-0.05, 0.05, -0.20, -0.05, 0.90, 0.12, 0.15, 0.05, 0.03}. The peak distortion value of 0.778 indicates a closed eye once again. Unlike the previous channels, this one has a relatively large number of pre-cursors as well as a phase characteristic that is quite nonlinear.

For an approximation to a noise-free channel the SNR was programmed for 100 dB as computed on a per-bit basis, i.e., $E_b/N_o = 1/\sigma^2$. A single unit pulse was transmitted and the MSE for this pulse was computed in accordance with the error criterion (2.3). Simulation runs were made to study the effects of the following parameters on equalizer

performance: the delay (D) between the arrival at the equalizer of the first precursor of the channel unit-pulse response and the occurrence of the locally generated reference pulse, the number of shift-register stages (N) in each filter-section, and the number of filter-sections (M) each of which has an associated tap-gain multiplier c_m. It will also prove convenient to introduce the notation (M,N,D) to describe the equalizer defined by these three parameters. For the study of D two different sets of values were used for the parameters M and N. In order to obtain as wide a range of D as possible the relatively large values of 32 and 20 were initially used for M and N, respectively. The second set of values was selected so that $N = \hat{L}$, the length of the unit-pulse response of the particular channel considered, and M was then set to the smallest multiple of four consistent with the constraint (2.27) i.e., $M \geq N + \hat{L} - 1$. Thus a comparison is obtained between relatively large and small equalizer structures. For the study of N, M was again set to 32 and D assigned a nominal value suitable for the channel considered. Selection of this value will be discussed shortly. In studying M, N was assigned the largest value consistent with (2.27) and D was again set to some nominal value. Finally, the equalized unit-pulse response of all three channels was determined using (32,20,D) and $(M,\hat{L},D)$ equalizers with M the smallest feasible value and with D set to the corresponding optimal value.

Selection of the nominal value of D involves several considerations. The basic convolution relation describing the channel is obtained from (2.26), which for the noise-free case and filter-section m reduces to

$$v_{mk} = \sum_{n=0}^{k} f_{mn}\, g_{k-n} = \sum_{n=0}^{k} f_{m,k-n}\, g_n, \qquad (2.39)$$

where f_{mk} is defined to be zero if $k \geq N$, and $\{g_m\}$ is the channel unit-pulse response. If there are i precursors $\{g_0, g_1, \ldots g_{i-1}\}$ then (2.39) implies that the main pulse g_i corresponding to a binary one cannot affect the equalizer output before i shifts of the N-stage register. It is intuitive to expect the best equalization to occur when both precursors and postcursors are able to influence the equalizer output sample corresponding to the transmitted unit pulse. However, since the MSE (2.3) depends on the filter-section outputs at each of the N+L-1 sample times required for the channel output sequence to pass completely through the equalizer, this may not always be true. In general the optimal delay can be expected to be somewhat larger than i, provided the shift-register length is on the order of 2i or greater. For shorter lengths the optimal delay may be less than i in order to retain a sufficient number of precursors in the shift register for the best equalizer output. Since the equalizer tap-gain

settings depend on D and the output depends simultaneously on all tap-gain multiplier outputs, the filter-sections cannot be treated independently in determining the optimal D. Consequently, the optimal D must be determined by trial and error, which is not feasible to incorporate in the initializing algorithm considered here; therefore, a nominal delay must be selected for a given type of channel.

Additional insights into the selection of the nominal delay can be gained by considering the frequency domain. Since each filter-section is a transversal digital filter with, in general, no symmetry in the set of multipliers $\{f_{mn}\}$, the phase vs. frequency characteristic is nonlinear and the corresponding delay cannot be expected to be constant. Rather, different frequencies will be delayed by different amounts. This same nonlinear phase characteristic is also typical of analog communication channels where it produces delay distortion. For example, the voice-grade telephone lines commonly used in data transmission typically employ such phase-distorting devices as loading coils and bandpass filters. Delay distortion tends to cause relatively long precursors in addition to an average time delay of the signal maximum, and also may cause a skewing of the signal peak which results in either a small delay or advance of the signal maximum from the average time delay.

The transmitted signal also undergoes amplitude distortion in the channel. This effect is due to nonuniform amplitude vs. frequency characteristics in both the channel passband and stopbands, and it causes dispersion of the transmitted pulse with the resulting precursors and post-cursors. Based on these considerations the nominal value of D for the LPF channel was taken to be approximately 0.25 N since the channel has a nearly linear phase characteristic over most of the passband. A nominal value of approximately 0.50 N was chosen for the VSB channel in view of the loading and modulation, whereas, for the BPF channel, where the degree of phase nonlinearity is somewhere between the other channels, simulation runs were made using each of the previous choices.

Results of the study of parameters D, M and N for the three aforementioned channels are presented in Figs. 2.4, 2.5 and 2.6. Corresponding numerical data is found in Tables 2.1 and 2.2. The effect on MSE when the delay D is varied while holding M and N fixed is shown in Part (a) of each figure. The anticipated variation of delay with the channel phase is apparent, and it can also be seen that the nominal values of D are reasonable.

It is interesting to note that for the LPF channel with M set to 32 any value of delay less than six essentially minimizes the MSE. Because of the large

amplitudes of the first two samples of the channel output, either could be considered to correspond to the transmitted unit pulse. Hence the channel delay is small and could easily be zero, depending on which sample the equalizer chooses to correspond to the unity output. Due to the nature of the particular filter-section tap-gains computed for this channel, the smallest residual MSE values occur when the channel output sequence begins to enter the shift register, rather than after most of the postcursors are included as might intuitively be expected.

The variation in performance with shift-register length (N) is shown in Part (b) of Figs. 2.4 through 2.6. Larger values of N produce better performance, as can be clearly seen. For the BPF channel there is little difference in the results for nominal delays of approximately 0.25 N and 0.50 N so long as N is larger than the channel unit-pulse response length ($\hat{L}$) of 11. However, when N becomes less than $\hat{L}$, the value of 0.50 N is clearly superior. This is as predicted, since the short shift-register lengths cannot hold complete information on both precursors and postcursors, and a delay of 0.25 N results in a relatively large absence of postcursors. The improvement in performance with an increasing number of filter-sections (M) is also clearly seen from Part (c) of each figure.

The unit-pulse response of the respective channel is shown in Part (d) of each figure both before equalization and for the smaller equalizer case. Thus the plotted unit-pulse response represents a relatively small equalizer structure. Although the resulting distortion is quite small in all cases, it can be further reduced by increasing the equalizer size. This can clearly be seen from Table 2.3 which gives the equalized responses for both large and small size equalizers and the optimal values of D pertinent to each case. From the table it is also apparent that the largest deviation of the equalized response from the desired values is about 0.08 for the BPF channel, while the corresponding figures for the LPF and VSB channels are on the order of 0.005 and 0.015 respectively.

The equalizer design developed in this chapter affords considerable reduction of ISI for the noise-free channels considered. All three channels were successfully equalized, even when N was less than $\hat{L}$, the channel unit-pulse response. This is an important observation because, as discussed in C.3, the amount of computation required is proportional to N^3. The effects of background noise and channel parameter variations will be analyzed in Chapter 3.

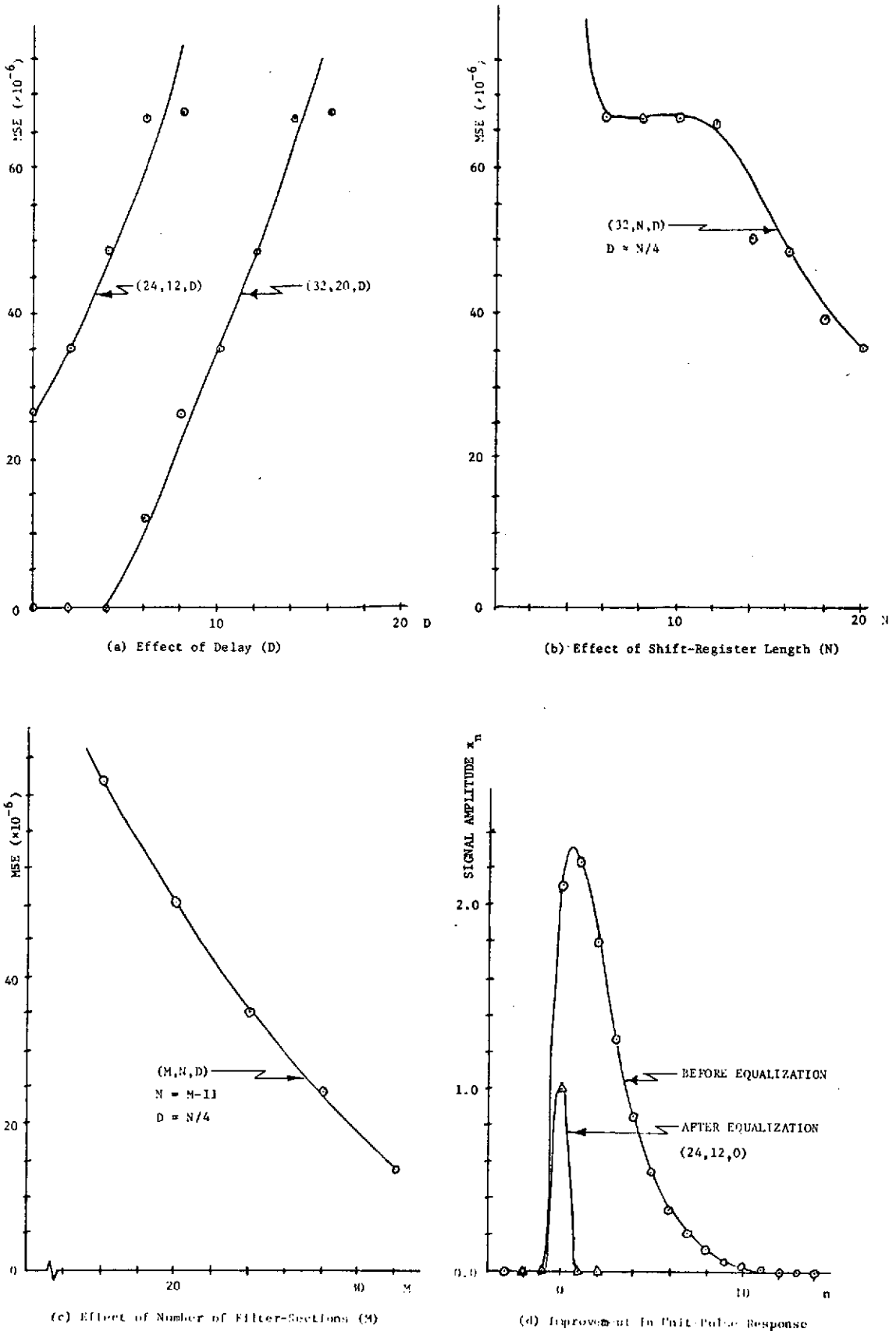

Fig. 2.4 Noise-Free Equalizer Performance for LPF Channel

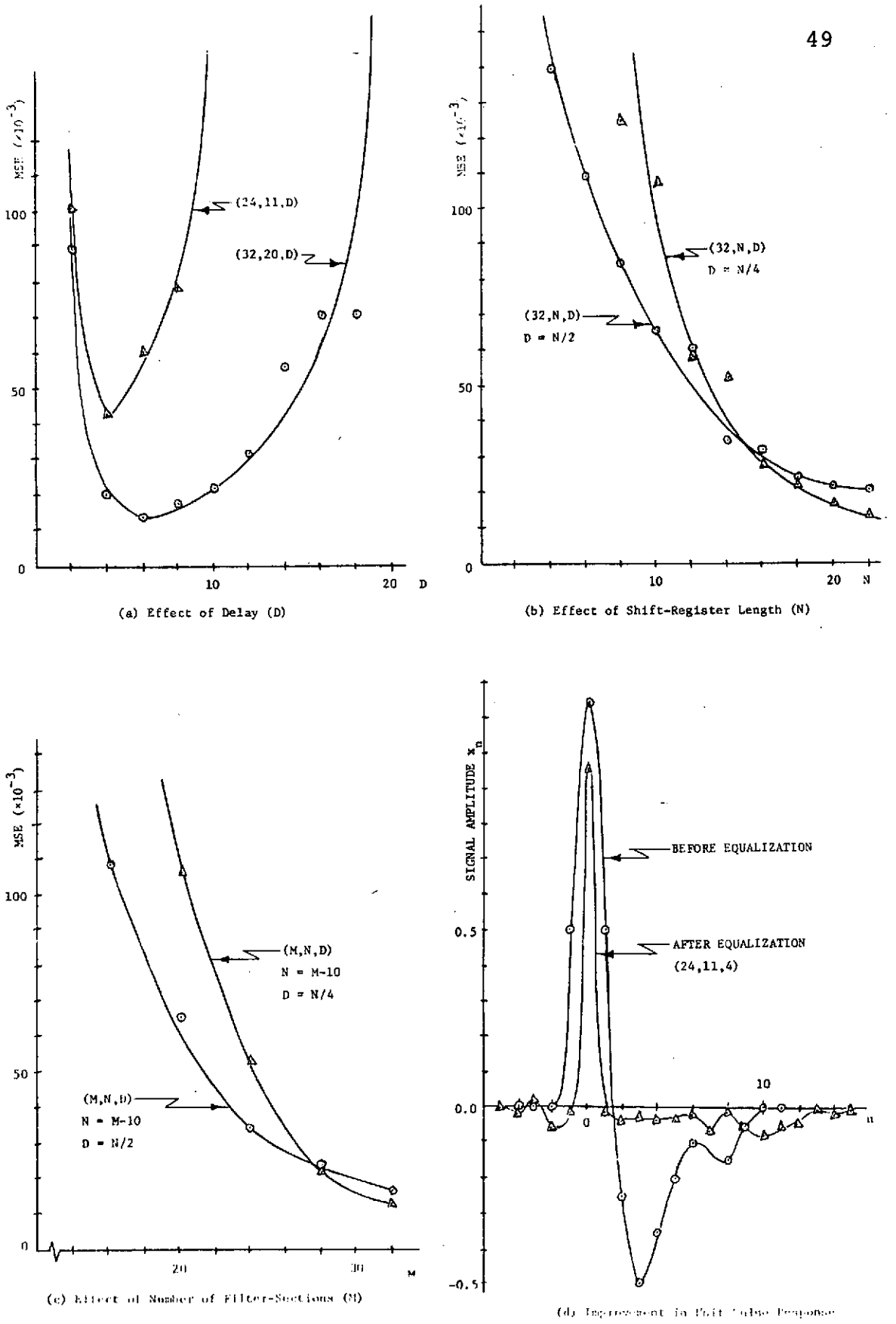

Fig. 2.5 Noise-Free Equalizer Performance for BPF Channel

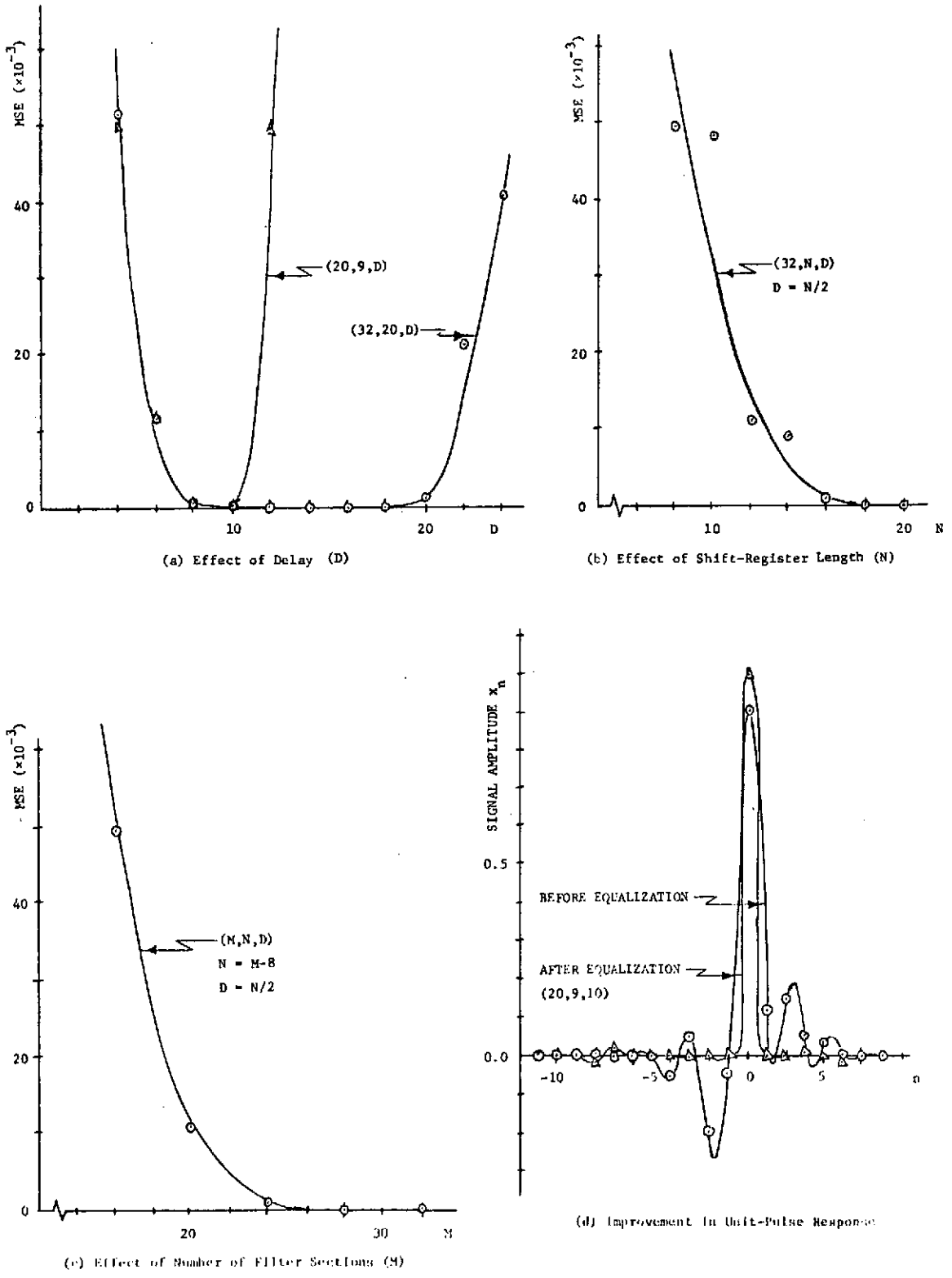

Fig. 2.6 Noise-Free Equalizer Performance for VSB Channel

TABLE 2.1

EFFECT OF DELAY (D) ON MSE FOR NOISE-FREE CHANNELS

LPF CHANNEL		
M	32	24
N	20	12
D	MSE($\times 10^{-6}$)	
0	0.00	26.67
2	0.01	35.31
4	0.01	48.54
6	12.43	67.38
8	26.27	67.52
10	35.31	$>10^4$
12	48.54	
14	67.38	
16	67.52	
18	$>10^4$	

BPF CHANNEL		
M	32	24
N	20	11
D	MSE($\times 10^{-3}$)	
2	88.26	100.90
4	20.06	42.39
6	13.85	60.04
8	17.27	75.73
10	21.28	204.40
12	31.40	
14	55.77	
16	71.09	
18	72.05	
20	371.70	

VSB CHANNEL		
M	32	20
N	20	9
D	MSE($\times 10^{-3}$)	
4	50.26	49.84
6	11.19	11.19
8	1.54	1.75
10	0.30	0.57
12	0.05	49.59
14	0.01	
16	0.00	
18	0.01	
20	0.22	
22	22.27	

TABLE 2.2

EFFECT OF SHIFT-REGISTER LENGTH (N) AND NUMBER OF FILTER-SECTIONS (M) ON MSE FOR NOISE-FREE CHANNELS

	CHANNEL			
	LPF	BPF	BPF	VSB
M	32	32	32	32
D	0.25N	0.25N	0.50N	0.50N
N	MSE			
	($\times 10^{-6}$)	← – – ($\times 10^{-3}$) – – →		
4	$>10^4$	322.90	139.50	
6	67.23	229.29	108.80	
8	67.03	124.50	83.76	49.84
10	67.14	107.30	65.49	48.16
12	66.88	57.50	60.02	11.19
14	50.40	52.71	34.12	9.25
16	48.35	26.77	32.35	1.54
18	39.23	22.37	23.98	1.28
20	35.31	16.83	21.82	0.30
22		13.68	20.31	0.20
24				0.05

	CHANNEL			
	LPF	BPF	BPF	VSB
N	M-11	M-10	M-10	M-8
D	0.25N	0.25N	0.50N	0.50N
M	MSE			
	($\times 10^{-6}$)	← – – ($\times 10^{-3}$) – – →		
16	67.07	229.20	108.80	49.85
20	50.10	107.3	65.52	11.19
24	35.06	52.71	34.14	1.52
28	24.85	22.37	23.97	0.30
32	13.42	13.68	17.02	0.05

TABLE 2.3

UNIT-PULSE RESPONSE WITH (d_n) AND WITHOUT (x_n) EQUALIZATION

LPF CHANNEL		
M	32	24
N	20	12
D	0	0
x_n	$d_n(\times 10^{-3})$	
0.00	0.0	0.0
2.11	1000.0	1000.0
1.80	0.0	0.0
1.28	.	.
0.85	.	.
0.55	.	.
0.34		
0.21		
0.12		
0.07		.
0.04		.
0.02		0.0
0.00		-3.6
.		3.7
.		0.6
.		0.0
		.
		.
		.

BPF CHANNEL		
M	32	24
N	20	11
D	6	4
x_n	$d_n(\times 10^{-3})$	
	0.0	
	-34.6	
	5.5	0.0
	-19.1	-92.6
	-5.3	19.9
0.00	-14.5	-50.6
0.50	-9.9	-7.0
1.15	986.2	957.7
0.50	-12.3	-13.2
-0.25	-14.8	-42.6
-0.50	-14.3	-24.5
-0.35	-15.7	-39.7
-0.20	-16.4	-32.2
-0.10	-17.1	-20.3
-0.15	-18.7	-76.7
-0.05	-18.0	-12.6
0.00	-22.3	-65.3
.	-21.5	-80.9
.	-22.5	-57.7
.	-21.6	-52.1
	-25.0	-6.6
	-28.7	-23.5
	-16.2	-7.8
	-44.2	-9.3
	-48.1	0.0
	-29.5	.
	-26.3	.
	-7.3	.
	-10.9	
	-4.2	
	-3.8	
	0.0	
	.	
	.	
	.	

VSB CHANNEL		
M	32	20
N	20	9
D	16	10
x_n	$d_n(\times 10^{-3})$	
	0.0	
	-0.8	
	10.0	
	-0.2	
	0.3	0.0
	-0.1	-1.6
	0.1	2.7
	0.0	-12.6
	0.1	10.5
	0.0	-2.6
0.00	0.0	3.1
-0.05	-0.1	-1.8
0.05	-0.1	1.4
-0.20	-0.1	-0.6
-0.05	0.1	0.5
0.90	1000.0	999.4
0.12	0.0	-0.8
0.15	0.0	-0.4
0.05	0.1	8.9
0.03	0.0	-0.8
0.00	0.0	-12.8
.	-0.1	-4.6
.	-0.1	0.0
.	-0.9	.
	-0.1	.
	0.0	.
	-0.1	
	0.0	
	.	
	.	
	.	

CHAPTER 3

ON-LINE OPERATION OF THE INITIALIZED EQUALIZER WITH NOISY CHANNELS

In this chapter the generalized transversal equalizer design of Chapter 2 is modified so that near-optimal operation is maintained on noisy channels during the message transmission which occurs after initialization. Since it is impractical to compute the exact gradient $\nabla J(\underline{c})$ of the error criteron (2.3) for noisy channels with ISI, a computationally simple method for estimating this quantity is presented. Some convergence properties of the resulting algorithm are then developed, and an eigenvalue condition is obtained which insures convergence in a stochastic sense under a stated set of assumptions. When these assumptions are violated the possibility of divergence exists, and a simple test is then proposed to determine when reinitialization is desirable. Finally, the effect of noise on both the initialization procedure and the on-line tracking is studied by means of a computer simulation using the same channel models as in Chapter 2.

3.1 THE NOISY ON-LINE ERROR CRITERION

Under the noise-free hypothesis of Chapter 2 the equalizer initialization procedure used a single isolated initializing pulse and there was only one iteration of the algorithm (2.2). It was feasible to compute the gradient of the error criterion exactly, at least to within the limitations of numerical accuracy, since all nonzero error samples ε_k resulting from the initializing pulse were available separately for the summation resulting from (2.3), i.e.,

$$\frac{1}{2} \nabla J(\underline{c}[0]) = \sum_{k=-\infty}^{\infty} \varepsilon_k \nabla \varepsilon_k \ .$$

However, when the equalizer is operated in a decision-directed mode with a noisy channel, then the situation is less straightforward. First, because of the channel noise the equalizer output sample is a discrete random variable. Consequently, the equalizer output and the error criterion are random as well, and some type of statistical averaging is indicated. Second, as the message pulses are no longer isolated at the channel output, but rather overlap to produce ISI, it is necessary to notationally distinguish between error samples corresponding to different transmitted pulses. Since the on-line equalizer will be adjusted once for each transmitted pulse, it will be convenient to

denote the error sample at iteration k by $\varepsilon[k]$. Notice that in general $\varepsilon[k]$ depends on the precursors and post-cursors of neighboring pulses as well as the k^{th} pulse. By taking the expected value of $\varepsilon^2[k]$ the error criterion $\bar{J}(\underline{c}[k])$ for the noisy on-line case, corresponding to (2.3), is given by

$$\bar{J}(\underline{c}[k]) = E\{\varepsilon^2[k]\}, \tag{3.1}$$

where E is the mathematical expectation operator.

The on-line algorithm for adjusting the equalizer tap-gains now becomes

$$\underline{c}[k+1] = \underline{c}[k] - \frac{1}{2}\gamma\, \nabla\bar{J}(\underline{c}[k]), \tag{3.2}$$

where γ is again the scalar convergence factor, and the gradient expression is

$$\frac{1}{2}\nabla\bar{J}(\underline{c}[k]) = \frac{1}{2}\nabla E\{\varepsilon^2[k]\}. \tag{3.3}$$

To simplify (3.3) the <u>filter-section output vector</u> $\underline{v}[k]$ is defined to be $\underline{v}[k] \triangleq (v_0[k], v_1[k], \ldots v_{M-1}[k])^T$ where the component $v_m[k]$ is the output value of the m^{th} filter-section at iteration k. Notice that $\underline{v}[k]$ is distinctly different from $\underline{v}_m$ defined in (2.28). If the decision-directed estimate d[k] of the k^{th} transmitted message pulse is denoted by $\hat{d}[k]$, then the error $\varepsilon[k]$ is given by

$$\varepsilon[k] = y[k] - \hat{d}[k] = \langle \underline{c}[k], \underline{v}[k] \rangle - \hat{d}[k],$$

and upon substitution into (3.3) there results

$$\begin{aligned} \frac{1}{2}\nabla E\{\varepsilon^2[k]\} &= E\{\varepsilon[k]\underline{v}[k]\} \\ &= E\{\underline{v}[k]\ \underline{v}^T[k]\ \underline{c}[k]\} - E\{\hat{d}[k]\underline{v}[k]\}. \end{aligned} \tag{3.4}$$

Although both variables $\varepsilon[k]$ and $\underline{v}[k]$ are explicitly available in the equalizer and their product is easily calculated, taking the expectation of the result on-line is not practical. To circumvent this difficulty the gradient (3.4) will be estimated using the so-called "noisy" estimate method suggested by Widrow [5]. The resulting expression is

$$\frac{1}{2}\ \nabla\bar{J}(\underline{c}[k]) \simeq \varepsilon[k]\ \underline{v}[k], \tag{3.5}$$

which is an unbiased estimate since

$$E\{\varepsilon[k]\ \underline{v}[k]\} = \frac{1}{2}\nabla\bar{J}(\underline{c}[k]).$$

The algorithm (3.2) is now approximated by the easily implementable form

$$\underline{c}[k+1] = \underline{c}[k] - \gamma\ \varepsilon[k]\ \underline{v}[k]. \tag{3.6}$$

Equation (3.6) involves no explicit a priori statistical information, but it is based on the decision-directed hypothesis that the binary sequence of equalizer

output decisions closely approximates the transmitted message sequence. It should also be noted that the initialization procedure requires a priori information in the form of the locally-generated initializing pulse replica. When a priori statistical information is available during on-line operation more accurate algorithms can be obtained, although the gradient determination generally involves more computation. Several such cases are discussed in B.7.

The equalizer structure developed in Chapter 2 can now be modified for on-line operation using (3.6). Fig. 3.1 shows the complete equalizer block diagram. All variables shown correspond to the output sample y[k], i.e., the delay caused by the channel and filter-sections is suppressed in the notation. The channel noise n[k] is added to the deterministic channel output s[k] to obtain the equalizer input x[k]. A filter-section output $v_m[k]$ is the weighted sum of those N input samples located in the shift register when y[k] occurs, and the weights are the filter-section tap-gains given by (2.37). The equalizer output is now given in vector form by

$$y[k] = \langle \underline{c}[k], \underline{v}[k] \rangle \quad .$$

The shift to on-line operation involves only two switchings as shown in Fig. 3.1. First, the equalizer output decisions rather than the reference generator

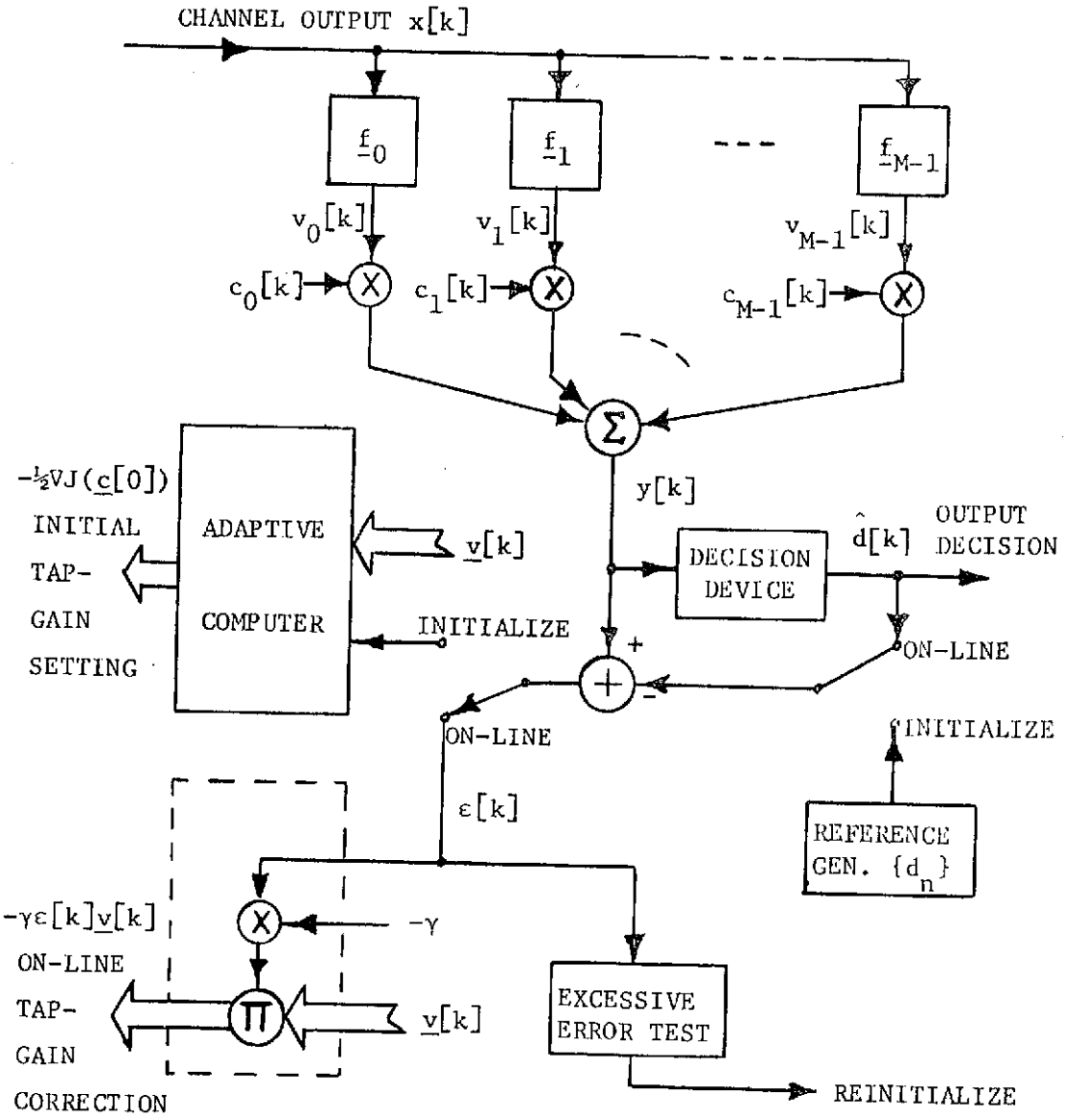

Fig. 3.1. Equalizer Structure With On-Line Tracking

output are fed back to produce the error signal in a decision-directed manner. Second, since the on-line computations are considerably simpler than those during initialization, it may be advantageous to use a set of digital multipliers to compute the on-line gradient estimate in lieu of the adaptive computer used for initialization. This is particularly true when the initialization computation is done by a time-shared general purpose computer. Of course, if the adaptive computer is continuously available the second switching operation is unnecessary.

A final design consideration involves the detection of erroneous operation. One simple method for doing this is to compare $|\varepsilon[k]|$ with some predetermined positive threshold value ε_{max} and to reinitialize the equalizer whenever ε_{max} is exceeded a given number of times in a specified period.

3.2 CONVERGENCE PROPERTIES OF THE ON-LINE ALGORITHM

Before investigating the convergence properties of the on-line algorithm (3.6), several preliminary considerations are necessary. Since the noise is random, this convergence must be considered in some statistical sense. It is also mathematically expedient to model the noise as a stationary random process with zero mean and finite

variance. Although some forms of background noise are not stationary, e.g., switching and impulse noise, other forms such as thermal noise do fit this model reasonably well, particularly if the time frame is not too long. Furthermore, if the equalizer setting deviates too much from the optimal setting, then the excessive-error test on the resulting errors $\varepsilon[k]$ will indicate the need for reinitialization.

Taking the expectation of (3.6) under the stated hypothesis and assuming $\underline{c}[k]$ is statistically independent of $\underline{v}[k]$ gives

$$\underline{c}[k+1] = \underline{c}[k] - \gamma\,(\underline{R}_{vv}\,\underline{c}[k] - \underline{r}_{dv}), \qquad (3.7)$$

where $\underline{R}_{vv} = \underline{R}_{vv}(0) \triangleq E\{\underline{v}[k]\,\underline{v}^T[k]\}$ is the filter-section correlation matrix, $\underline{r}_{dv}(n) \triangleq E\{\hat{d}[k]\,\underline{v}[k]\}$ defines a cross-correlation vector, and $\underline{c}[k]$ now denotes the expectation of the tap-gain vector. A channel output M-vector $\underline{x}[k]$ at iteration k consists of the current shift-register contents, which produce y[k], for the first N components, and the most recent M-N samples to have passed through the register as the remaining components. By now defining the crosscorrelation matrix $\underline{R}_{xx} = \underline{R}_{xx}(n) \triangleq E\{\underline{x}[k]\underline{x}^T[k]\}$ and vector $\underline{r}_{dx} \triangleq E\{\hat{d}[k]\underline{x}[k]\}$, and noting that $\underline{v}[k]=\underline{F}^T\,\underline{x}[k]$, it follows that $\underline{R}_{vv}=\underline{F}^T\,\underline{R}_{xx}\,\underline{F}$ and $\underline{r}_{dv} = \underline{F}^T\,\underline{r}_{dx}$. Under the mild assumption that the signal s[k] and noise n[k] are independent, it is shown in B.8 that $\underline{R}_{xx}$ is positive definite and thus nonsingular. It can then be shown by the same procedure as in B.4 that $\underline{r}_{dv}$

lies in the range space of $\underline{R}_{vv}$, and consequently (3.7) is convergent for $\hat{\underline{c}} = \underline{R}_{vv}^{-I}\ \underline{r}_{dv}$. Although this is not, of course, the exact optimal solution of the original deterministic algorithm (2.2), it is the optimal expected solution of the on-line algorithm (3.6) in which the "noisy" gradient estimates are used.

To study the convergence of (3.7) $\hat{\underline{c}}$ is first subtracted from both sides of the equation which is then written in terms of the tap-gain error vector $\underline{e}[k] = \underline{c}[k] - \hat{\underline{c}}$ as

$$\underline{e}[k] = (\underline{I} - \gamma\ \underline{R}_{vv})^{k-1}\ \underline{e}[1]. \tag{3.8}$$

Since $\underline{R}_{vv}$ is a correlation matrix it is symmetric and positive semidefinite. Consequently, all eigenvalues $\{\rho_i\}$ are both real and nonnegative and there exists an orthonormal set of eigenvectors $\{\underline{\zeta}_i\}$. Letting Θ and $\bar{\Theta}$ denote the index sets corresponding to the positive and zero eigenvalues of $\underline{R}_{vv}$, respectively, a unique orthogonal decomposition $\underline{e}[k] = \underline{e}^{+}[k] + \underline{e}^{o}[k]$ exists, so that $\langle \underline{e}^{+}[k],\ \underline{e}^{o}[k] \rangle = 0$. In this decomposition $\underline{e}[1]$ is written as

$$\underline{e}[1] = \underline{e}^{+}[1] + \underline{e}^{o}[1] = \sum_{i \in \Theta} \alpha_i \underline{\zeta}_i + \sum_{i \in \bar{\Theta}} \alpha_i \underline{\zeta}_i \tag{3.9}$$

where $\{\alpha_i\}$ are the appropriate coefficients for the basis $\{\underline{\zeta}_i\}$. Substitution of (3.9) into (3.8) now gives

$$\underline{e}[k] = \underline{e}^{+}[k] + \underline{e}^{o}[k] = \sum_{i\in\Theta} \alpha_i (1-\gamma\rho_i)^{k-1} \underline{\zeta}_i + \sum_{i\in\bar{\Theta}} \alpha_i \underline{\zeta}_i . \tag{3.10}$$

The last term of (3.10) is simply $\underline{e}^{o}[1]$ as can be seen from (3.9), i.e., $\underline{e}^{o}[k]$ will converge if the convergence factor γ is chosen to insure that

$$\max_{i\in\Theta} \{ | 1 - \gamma \rho_i | \} < 1, \tag{3.11}$$

since in this case

$$|| \underline{e}^{+}[k] || \leq \max_{i\in\Theta} \{|1-\gamma\rho_i|\} \, || \sum_{i\in\Theta} \alpha_i \underline{\zeta}_i || < || \underline{e}^{+}[1] || .$$

Therefore, the condition to insure convergence of (3.7) is found from (3.11) to be

$$\gamma < 2/\rho_{max} ,$$

where

$$\rho_{max} \geq \rho_i \text{ for all } i.$$

3.3 SIMULATION RESULTS FOR NOISY CHANNELS

To study the combined theory for equalizer initialization and on-line operation, the computer simulation used in Chapter 2 was expanded to reflect the theoretical results of this chapter. In particular, provisions were made for generating a random binary sequence to simulate the transmitted message, for adding gaussian noise with specified statistics, for adjusting the equalizer tap-gains at each bit time, and for detecting erroneous operation. The

computer simulation is described in some detail in C.1 and C.2.

The equalizer performance was again evaluated over the three channels discussed in 2.6, i.e., the LPF, BPF, and VSB channels, but the SNR was varied from the essentially noise-free value of 100 dB down to 15 dB. At 15 dB the noise standard deviation is 0.1778 and the probability of the noise alone causing a bit error is 0.0025. When this effect is superimposed on the distortion due to ISI, it is apparent that the equalizer alone should not be expected to perform well when the SNR is in this range. In practice such low SNR situations are commonly handled by coding, and are usually incompatible with the high baud rate to bandwidth ratios that justify sophisticated equalization.

In addition to the equalizer parameter set (M,N,D) defined in 2.6, the convergence factor γ must be specified for the on-line algorithm. It was found by trial and error that a good selection was 0.02, which represents 4% of the decision threshold value. Larger values of γ tend to increase the bit error rate when there is noise in the channel, and can lead to divergence if the increase is too large. On the other hand, smaller values result in slower equalizer convergence rates. The transmitted message was simulated by repeatedly generating a 255 bit pseudo-noise (PN) sequence, and the error criterion was taken to

be the average MSE $\varepsilon^2[k]$ after 500 iterations of the on-line algorithm. In determining $\varepsilon[k]$ the transmitted bit $d[k]$ was used in place of the output device decision $\hat{d}[k]$ for programming convenience. This substitution has no effect when there are no bit errors, and the effect is considered negligible for the SNR range considered. Data in which one or more isolated bit errors occurred is indicated by an asterisk in the appropriate figures.

The first parameter studied was the delay (D) between the arrival at the equalizer of the first precursor of the channel unit-pulse response and the occurrence of the locally-generated reference pulse for initialization. Since D depends on such nonlinear effects as the channel and filter-section phase characteristics, it is impractical to determine the optimal value analytically. As in 2.6, simulation runs were made for a relatively large equalizer structure, (32,20,D), and for one for which the equalizer shift-register length N equaled the channel unit-pulse response, i.e., the structure is $(M,\hat{L},D)$. The simulation results are shown in Part (a) of Figs. 3.2, 3.3, and 3.4, from which the effect of D as well as channel SNR at 20 dB and 40 dB can be seen. All three channels were equalized successfully for a SNR of 30 dB or greater using the larger equalizer structure. However, the smaller structure was unable to equalize the BPF channel, and the larger structure

was ineffective on this channel at 20 dB; consequently, a value of 30 dB vice 20 dB was used. The curves in Figs. 3.2 and 3.4 indicate that there is no significant improvement in performance for the larger structures provided D is properly chosen, and in fact for the VSB channel the smaller structure performs slightly better. However, there is a much wider range of feasible values of D for the larger structures. Thus the possibility of poor equalization due to an inaccurate choice of D with a small structure must be weighed against the complexity and cost of realizing a larger structure. Since the accurate selection of a nominal value of D depends on knowledge of the channel, a larger structure is particularly suitable when the channel is relatively unknown, or may be expected to change significantly after initialization. Based on the consideration of 2.6 and the experimental results, nominal values for D of 0.50 N, 0.25 N, and 0.80 N were selected for the LPF, BPF, and VSB channels, respectively.

The next parameter studied was the shift-register length (N). The result of varying N with M set to 32 and D to the corresponding nominal value is shown for each channel in Part (b) of Figs. 3.2, 3.3, and 3.4. For high SNR values performance tends to either improve or remain nearly constant with increasing N. However, with the LPF channel the performance degrades slightly with $N > 8$.

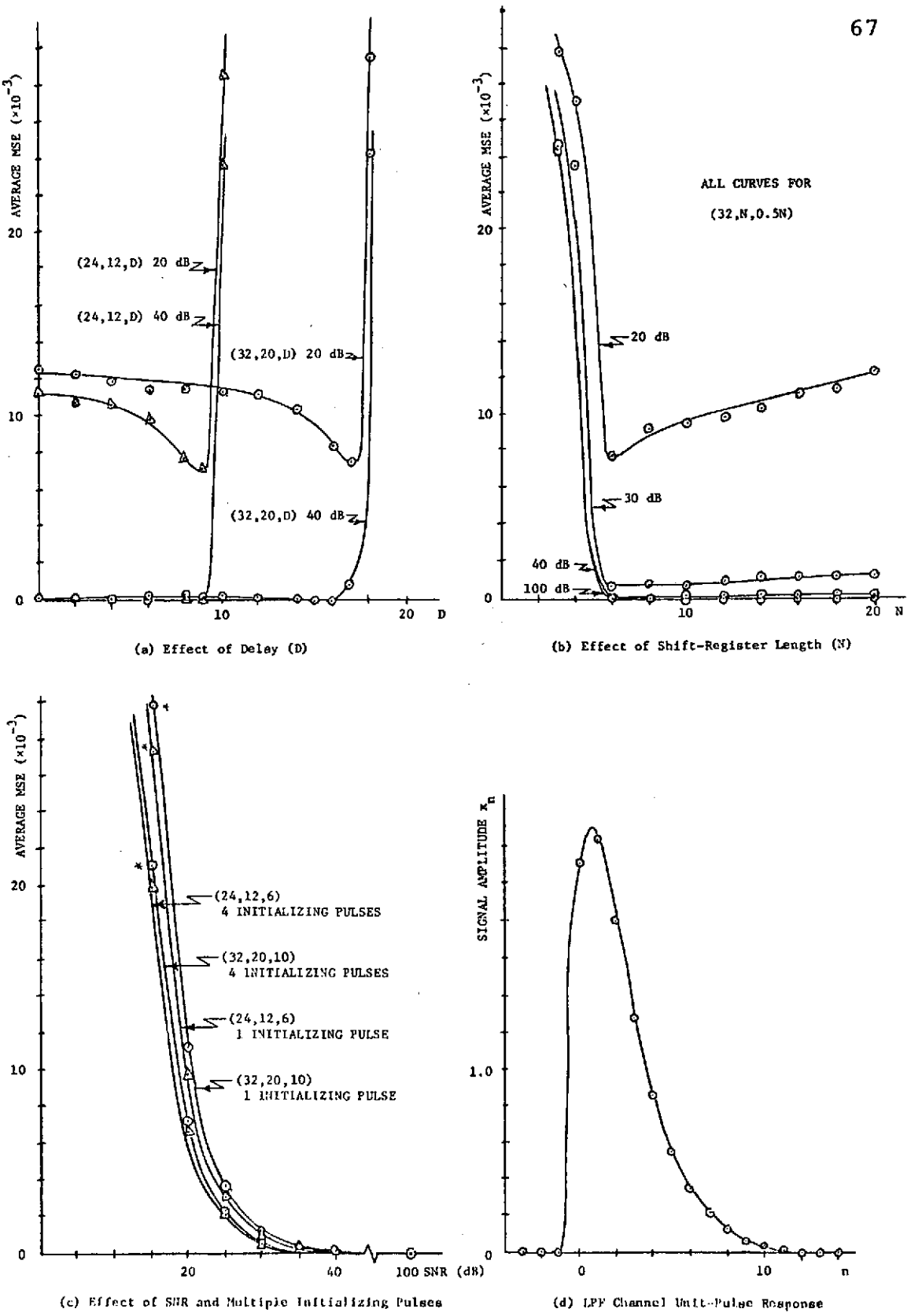

Fig. 3.2 Equalizer Performance for Noisy LPF Channel

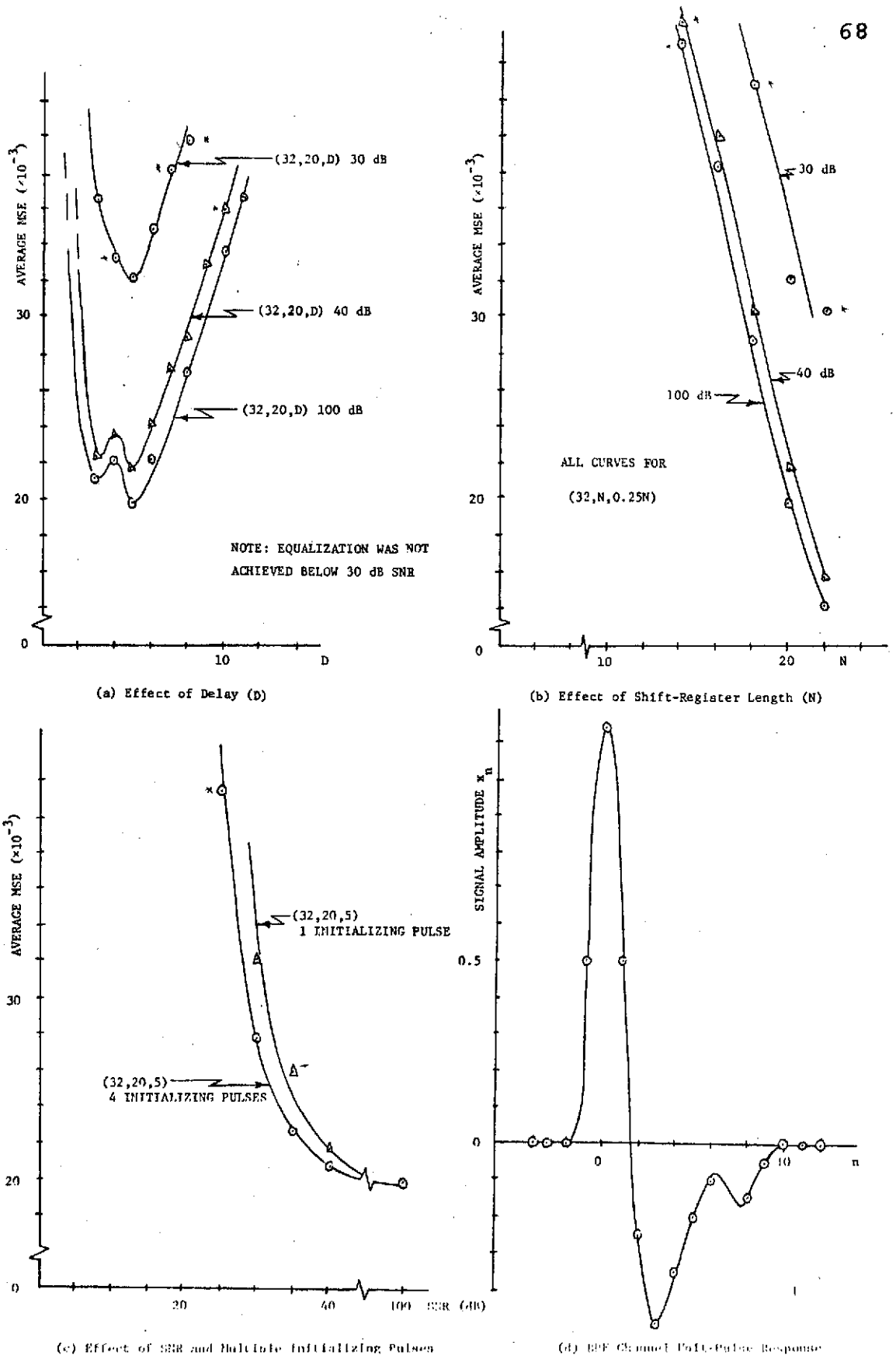

Fig. 3.3. Equalizer Performance for Noisy BPF Channel

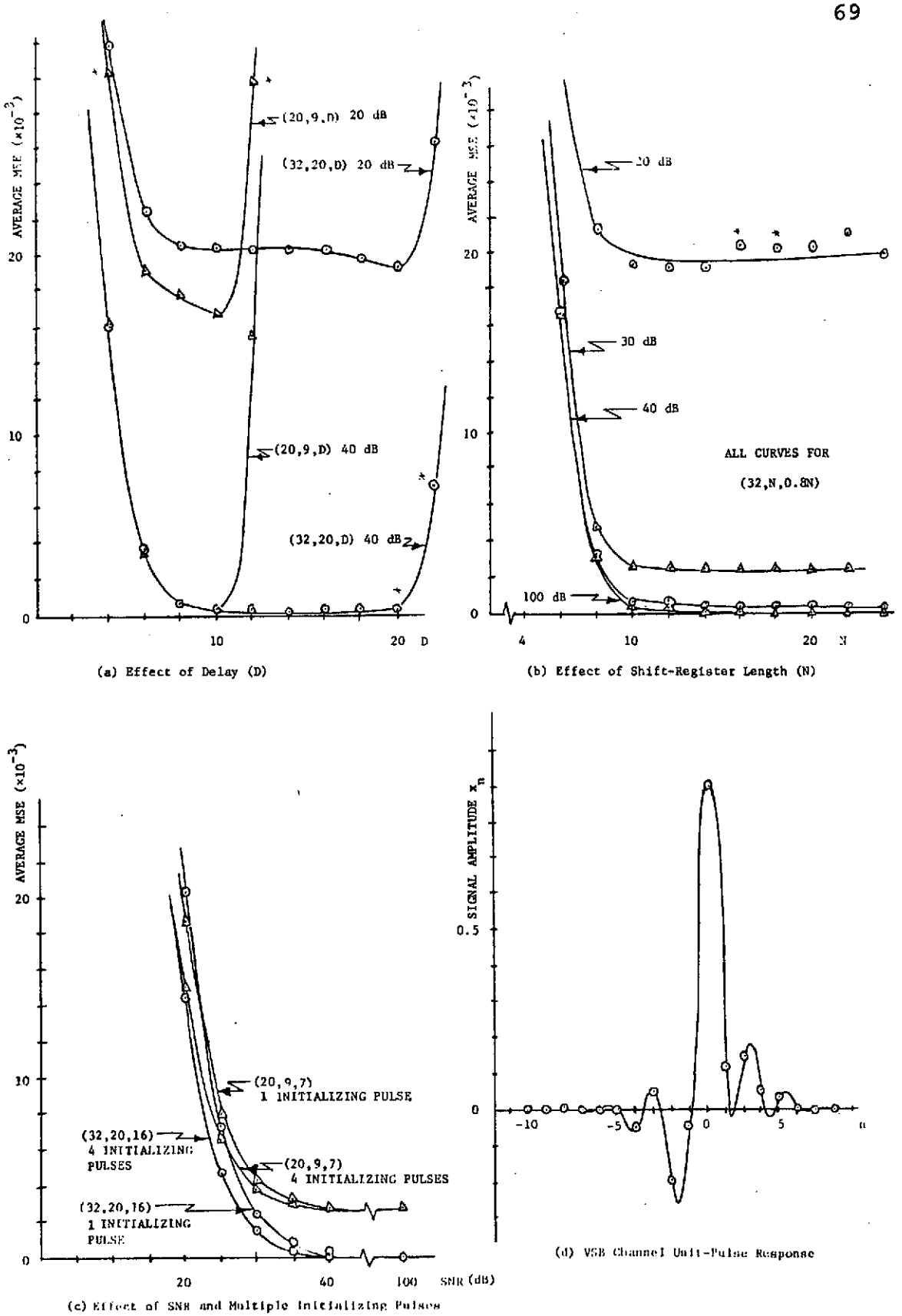

Fig. 3.4 Equalizer Performance for Noisy VSB Channel

This may be attributed in part to the increased number of arithmetic operations, each of which is reduced in accuracy by the noise, that is inherent in the larger structures. It is also evident for the LPF and VSB channels that, for SNR $\geq$ 30 dB, there is a lower value of N above which there is little change in performance. As previously noted, this is important since the amount of computation required for initialization is reduced considerably with N. Again, however, it should be kept in mind that small values of N make the selection of D more crucial. It is interesting to note that for the LPF channel equalization was achieved when the SNR was 20 dB with a shift-register having only three taps.

Part (c) of Figs. 3.2, 3.3, and 3.4 shows the effect of SNR on equalizer performance when both a single initializing pulse and an average of four such pulses were used by the equalizer to identify the channel during initialization. It is clear in each case that the use of multiple initializing pulses is quite benefitial where the SNR is low and the noise represents a significant percentage of the unit-pulse amplitude. This is not the case at SNR above 30 dB, so little advantage results from using multiple initializing pulses.

The LPF and VSB channels were successfully equalized with a 20 dB SNR; however, 30 dB was needed before the BPF

channel was equalized without bit errors. As can be seen from a comparison of the channel unit-pulse responses shown in Part (d) of Figs. 3.2, 3.3, and 3.4, the BPF channel has more overshoot than the other two, and also has a number of large, negative postcursors. Thus it is expected to be the more difficult channel to equalize. The fact that no value of N was reached above which the equalizer performance remained fairly constant suggests that better equalization is attainable for larger equalizer parameters than the maximum of (32,22,D) permitted by the particular computer simulation used. There are a number of tradeoffs in selecting the equalizer parameters which can now be summarized. First, selecting large values of M and N gives a larger range of values from which to select the nominal value of D, which is particularly important if little is known about the channel a priori. On the other hand, smaller values of N greatly reduce the amount and time of computation required for initialization, while smaller values of M reduce the number of on-line multiplications required at each algorithm iteration. These considerations can also be thought of as a tradeoff between the accuracy desired from the equalizer and the cost that must be paid to achieve it.

CHAPTER 4

CONCLUSIONS AND RECOMMENDATIONS

The design of a linear digital equalizer for reducing ISI on digital communication channels has been developed and evaluated. This design includes an initialization algorithm employing a digital computer, as described in Chapter 2, and a simplified on-line algorithm, developed in Chapter 3, which requires only a set of digital multipliers. The effects of the main equalizer parameters (M,N,D) and of channel SNR were studied by means of a digital computer simulation for three baseband channels. All channels were equalized successfully for a SNR of 30 dB or greater, and two were equalized at 20 dB. Considering a digital computer as a sophisticated digital filter, the equalizer simulation is quite realistic except for the relatively long word length of 36 bits.

There are a number of considerations to be taken into account when realizing an equalizer of the type presented here. First, the maximum shift-register length N is clearly limited by the digital computer used for the initialization computation. The amount and speed of digital hardware available for the shift register,

multipliers, and summers limits N, and M as well. Another important consideration is the determination of a nominal value for the delay D. Good estimates of D will generally require some a priori knowledge of the channel characteristics or some preliminary experimental data. An alternative procedure, which might be necessary with a completely unknown channel, is to iterate the initialization procedure several times using different values for D. If the nominal value of D is uncertain, or if considerable variation is expected in the channel unit-pulse response, then a large value of N will make the equalizer performance less sensitive to D. The nominal value of D ranged from 0.25N to 0.8N depending on the choice of channel model. A third design consideration is the test for excessive error indicated in Fig. 3.1. Since single bit errors are highly probable when the SNR is low or the channel badly equalized, some type of averaging is desirable to avoid an excessive number of reinitialization commands. This could be done, for example, by keeping a running count of the number of times $|\varepsilon[k]|$ exceeds a predetermined limit ε_{max} over the last m iterations and reinitializing only if it is exceeded $n < m$ times, where m and n are known integers. Such a scheme would also prevent short bursts of impulse or switching noise from causing a needless reinitialization. This scheme could be easily implemented with digital

hardware, as could more sophisticated schemes which might be devised for particular installations.

There are a number of interesting areas which could provide motivation for further investigation. One particularly germane area is the equalizer performance on time-varying channels and the effect of the convergence factor γ on this tracking. The computer simulation of App. C is easily modified for such a study. A second more theoretical area involves reduction of the amount of computation required for initialization. More efficient algorithms for finding eigenvalues and eigenvectors could perhaps be devised for particular equalizer installations. The use of a suboptimal procedure to estimate the initial equalizer settings might also reduce the initialization time considerably. The third area to be suggested concerns the effects of short word lengths and the related problems of truncation and round-off error. There has been some interesting theoretical work in this area [13], and experimental simulations could be done using a minicomputer for which a typical word length is 16 bits. A fourth area for further investigation is the possibility of further simplifying the on-line algorithm by using only polarity information of some of the variables. This type of scheme has been called "hybrid" by Hirsch and Wolf [10] who studied it for transversal equalizers.

A final comment concerns the feasibility of physically realizing an equalizer of the type developed herein. Although the equalizer requires a considerable amount of computing equipment for initialization, the constant improvement in integrated circuit technology and the development of faster, more complex components make the production of such equipment appear feasible. Furthermore, the proliferation of time-sharing systems and inexpensive minicomputers makes large amounts of computing facility widely available. Thus the equalizer design presented herein is both theoretically and physically feasible.

APPENDIX A

MATHEMATICAL FUNDAMENTALS

This appendix will provide both review and supplemental mathematical material. It deals first with selected topics from linear algebra and operator theory on finite dimensional inner product spaces. This theory is then specialized to matrices on real and complex vector spaces, with the subject of pseudoinverse matrices covered in a separate section. Finally, the classical z-transform is briefly discussed and some useful properties are derived.

A.1 LINEAR VECTOR SPACES AND OPERATORS

A mathematical field F consists of a set of elements called scalars which satisfy a particular set of axioms [35]. The only fields used herein are the sets of all real and all complex numbers. A linear vector space over a field F is denoted (U,F) or just U when F is clearly understood. The elements of U, called vectors, also satisfy a certain set of axioms [36] and are related to each other and to the scalars by the operations of vector addition and scalar multiplication. Linear vector spaces are also called

vector spaces or linear spaces. Vector spaces are called real or complex according to whether the field is real or complex, respectively. A set of vectors $\{\underline{x}_1, \underline{x}_2, \ldots \underline{x}_K\}$ in U spans U if every vector $\underline{y}$ in U can be expressed as a linear combination of the set, i.e., if there exists a set of scalars $\{\alpha_1, \alpha_2, \ldots \alpha_K\}$ such that

$$\underline{Y} = \sum_{k=1}^{K} \alpha_k \underline{x}_k \ .$$

A set $\{\underline{x}_m\}$ is linearly independent if the only way that their linear combination can be made to equal zero is for all of the scalar multipliers to be zero. Any linearly independent set that spans U is called a basis of U, and the number of elements in any basis is called the dimension of U. Only vector spaces of finite dimension will be considered here.

Let (U, F) and (V, F) be vector spaces with α, β being arbitrary scalars in F and $\underline{w}$, $\underline{x}$, $\underline{y}$ arbitrary vectors in U. Suppose that to each element of U there is assigned a unique element of V, then the collection of these assignments is called a map T from U into V. Maps are also called functions, transformations, or operators, and these terms will be used interchangeably.

The identity operator I is defined by $I\,\underline{x} = \underline{x}$, and a map T^{-1} satisfying the relation $T^{-1}T = T\,T^{-1} = I$ is called the inverse of T. The inverse operator may not exist; however, when it does then T is said to be invertible. The set of

scalars λ for which the map $(T - \lambda I)$ is not invertible is called the spectrum of T and will be denoted $\sigma[T]$. When T satisfies the relation

$$T(\alpha \underline{x} + \beta \underline{y}) = \alpha T(\underline{x}) + \beta T(\underline{y})$$

it is called a linear operator. It is interesting to note that a field is a vector space, and also the set of all linear operators from U into V forms a vector space in which each linear operator is itself a vector space [37]!

There are many important maps between vector spaces. An inner or scalar product is a scalar-valued function of two vectors, denoted $\langle \underline{x}, \underline{y} \rangle$, which is defined by the following three axioms;

i. $\langle \underline{y}, \underline{x} \rangle = \langle \underline{x}, \underline{y} \rangle^*$

ii. $\langle \alpha \underline{x} + \beta \underline{y}, \underline{w} \rangle = \alpha \langle \underline{x}, \underline{w} \rangle + \beta \langle \underline{y}, \underline{w} \rangle$

iii. $\langle \underline{x}, \underline{x} \rangle \geq 0$ with equality only when $\underline{x} = \underline{0}$.

A vector space with an inner product defined on it is called an inner product space. The inner product determines the relative directions of vectors, and two vectors are orthogonal if $\langle \underline{x}, \underline{y} \rangle = 0$. The notion of distance in vector spaces is indicated by a nonnegative scalar-valued map called a norm, $\| \underline{x} \|$, which is also defined axiomatically by

i. $\| \alpha \underline{x} \| = |\alpha| \cdot \| \underline{x} \|$

ii. $\| \underline{x} + \underline{y} \| \leq \| \underline{x} \| + \| \underline{y} \|$

iii. $\| \underline{x} \| \geq 0$ with equality only when $\underline{x} = \underline{0}$.

A set of vectors $\{\underline{x}_m\}$ is an orthonormal set if $\| \underline{x}_m \| = 1$ for any m and any two distinct vectors are orthogonal. A vector space with a norm defined on it is a normed vector space. For the finite dimensional vector spaces considered herein it will be convenient to define the inner product of two vectors, say $\underline{x} = (x_1, x_2, \ldots x_K)^T$ and $\underline{y} = (y_1, y_2, \ldots y_K)^T$, by

$$\langle \underline{x}, \underline{y} \rangle \triangleq \sum_{k=1}^{K} x_k y_k^* = \underline{x}^T \underline{y}^* ,$$

which then defines a norm of $\underline{x}$ by

$$\| \underline{x} \| \triangleq \sqrt{\langle \underline{x}, \underline{x} \rangle} .$$

A map T, not necessarily linear, is called convex if for every value of α satisfying $0 \leq \alpha \leq 1$, it has the following property;

$$T([1 - \alpha] \underline{x} + \alpha \underline{y}) \leq (1 - \alpha) T(\underline{x}) + \alpha T(\underline{y}).$$

If the inequality is strict, then T is strictly convex.

For every linear operator T on an inner product space U there exists another operator T^A called the adjoint of T defined by

$$\langle T(\underline{x}), \underline{y} \rangle = \langle \underline{x}, T^A(\underline{y}) \rangle .$$

If $T = T^A$ then T is called a self-adjoint operator, and if $T T^A = T^A T$ then T is a normal operator. A projection operator is one which is both self-adjoint and idempotent,

i.e., $T\ T = T$. Any map for which the adjoint equals the inverse is a unitary operator.

A final important notion deals with the geometric interpretation of a map T from U to V. The set of elements $\underline{x}$ in U such that $T\ \underline{x} = \underline{0}$ is called the null space, or sometimes the Kernel, of T and is denoted $N(T)$. The set of elements $\underline{y}$ in V which can be written $\underline{y} = T\ \underline{x}$ for some $\underline{x}$ in U is called the range space, or image, of T and is denoted $R(T)$. The null space of T is related to the range space of its adjoint T^A in the following way: every element in $N(T)$ is orthogonal to every element of $R(T^A)$. To show this note first that $\underline{x}$ belongs to $N(T)$ if and only if $\langle \underline{y},\ T\underline{x} \rangle = \underline{0}$ for any $\underline{y}$ in U, and since $\langle \underline{y},\ T\underline{x} \rangle = \langle T^A \underline{x},\ \underline{y} \rangle$ then $\underline{x}$ is orthogonal to every $\underline{w}$ in U such that $\underline{w} = T^A\ \underline{y}$ for some $\underline{y}$. But this set of $\underline{w}$'s is just the range space of T^A, hence the desired result is proven.

A useful consequence of this last result is that the range spaces $R(T^A)$ and $R(T^A\ T)$ are identical. This can be proved indirectly by first noting that if $\underline{x}$ is in $N(T^A)$ then $(T\ T^A)\ \underline{x} = T(T^A \underline{x}) = T(\underline{0}) = \underline{0}$ so $\underline{x}$ is also in $N(T^A\ T)$. Conversely, if $\underline{x}$ is in $N(T^A\ T)$ then $\| T\underline{x} \|^2 = \langle T\underline{x},\ T\underline{x} \rangle = \langle \underline{x},\ T^A\ T\underline{x} \rangle = \langle \underline{x}, \underline{0} \rangle = \underline{0}$, so that $T\underline{x} = \underline{0}$ and $\underline{x}$ is in $N(T)$. The proof is completed by noting that every element of $R(T^A)$ and $R(T^A\ T)$ is orthogonal to every element of $N(T)$ and

$N(T^A T)$, respectively. Hence $R(T^A)$ and $R(T^A T)$ must then be identical.

A.2 MATRICES

An M x N <u>matrix</u> $\underline{T}$ is a rectangular array of elements having M rows and N columns; the element in row m and column n is denoted by t_{mn} and the entire matrix $\underline{T}$ is sometimes denoted by $[t_{mn}]$. By defining the sum of two matrices and the multiplication of a matrix by a scalar in the usual way, it can be shown that the set of all M x N matrices, with elements in the same field F, forms a vector space [38]. The way that matrices are used to represent linear operators is made precise by the following <u>Representation Theorem</u> [39]:

> Let (U, F) and (V, F) be real or complex vector spaces with bases $\{\underline{u}_1, \underline{u}_2, \ldots \underline{u}_N\}$ and $\{\underline{v}_1, \underline{v}_2, \ldots \underline{v}_M\}$, respectively, and let T be a linear map from U to V. Then, with respect to these bases, T is represented by the M x N matrix $\underline{T} = [t_{mn}]$ where the elements of column k of $\underline{T}$ are the components of $T(\underline{u}_k)$ with respect to the basis $\{\underline{v}_m\}$.

When M = N then $\underline{T}$ is a square matrix and U is identical to V, i.e., T maps U into itself. If M = N and $\{\tilde{\underline{u}}_1, \tilde{\underline{u}}_2, \ldots \tilde{\underline{u}}_N\}$ is another basis for U in which $\tilde{\underline{T}}$ represents T, then there exists a nonsingular square matrix $\underline{P}$ that

relates these representations according to $\underline{\tilde{T}} = \underline{P}^{-1}\ \underline{T}\ \underline{P}$. Such a mapping is called a similarity transformation, and the matrices $\underline{T}$ and $\underline{\tilde{T}}$ are said to be similar. Thus similar matrices represent the same operator in different bases.

For a square matrix $\underline{T}$ of order N the spectrum $\sigma[\underline{T}]$ is that set of scalars for which $(\underline{T} - \lambda\ \underline{I})$ fails to be invertable, or equivalently, for which the determinant $\det(\underline{T} - \lambda\ \underline{I}) = 0$. This last equation is a polynomial in λ of order N, so that $\sigma[\underline{T}]$ contains exactly N elements which may not be distinct. These elements are called the eigenvalues of $\underline{T}$, or sometimes proper or characteristic values. The spectrum is a property of the abstract operator T and does not depend on the particular representation, e.g., $\underline{T}$ or $\underline{\tilde{T}}$. This can be seen by recalling that $\underline{\tilde{T}} = \underline{P}^{-1}\ \underline{T}\ \underline{P}$ so that

$$\det(\underline{\tilde{T}} - \lambda\ \underline{I}) = \det(\underline{P}^{-1}\ \underline{T}\ \underline{P} - \lambda\ \underline{P}^{-1}\ \underline{P})$$

$$= \det(\underline{P}^{-1})\ \det(\underline{P})\ \det(\underline{T} - \lambda\ \underline{I}) = \det(\underline{T} - \lambda\ \underline{I}).$$

Any nonzero vector $\underline{p}$ such that $(\underline{T} - \lambda\ \underline{I})\ \underline{p} = \underline{0}$, with λ in $\sigma[T]$, is called an eigenvector of $\underline{T}$ corresponding to the eigenvalue λ.

There are a number of special matrices that will prove useful herein, some of which represent the various operators defined in Section A.1. The square matrix representing the adjoint T^A of an abstract linear transformation T on vector space U is the complex conjugate

transpose of $\underline{T}$ denoted $\underline{T}^H$ as can be seen from the following,

$$\langle \underline{x}, \underline{T}\,\underline{y} \rangle = \underline{x}^T\,(\underline{T}\,\underline{x})^* \quad = \underline{x}^T\,\underline{T}^*\,\underline{y}^* = \langle \underline{T}^H\,\underline{x}, \underline{y} \rangle .$$

When T is self-adjoint $\underline{T} = \underline{T}^H$ and $\underline{T}$ is called a hermitian matrix. If the vector space U is real rather than complex, then $\underline{T}^H = \underline{T}^T$ and T being self-adjoint implies $\underline{T} = \underline{T}^T$. Such a matrix is called real symmetric. The eigenvalues of an N x N hermitian matrix $\underline{T}$ are always real, and there always exists a set $\{\underline{p}_m\}$ of N linearly independent eigenvectors which are orthogonal [40]. In fact, since when $\underline{p}$ is an eigenvector then $\underline{p}/\,\|\underline{p}\|$ is too, then the set $\{\underline{p}_m\}$ can always be taken orthonormal. Writing the eigenvector equation $\underline{T}\underline{p}_m = \lambda_m\,\underline{p}_m$ for m = 1, 2,...N in matrix form gives

$$\underline{T}[\underline{p}_1, \underline{p}_2, \cdots \underline{p}_N] = [\underline{p}_1, \underline{p}_2, \ldots \underline{p}_N]\ \mathrm{diag}(\lambda_1, \lambda_2, \ldots \lambda_N)$$

or, more concisely in terms of the matrices $\underline{P}$ and $\underline{\Lambda}$,

$$\underline{T}\,\underline{P} = \underline{P}\,\underline{\Lambda}.$$

Furthermore, since $\langle \underline{p}_m, \underline{p}_n \rangle = \delta_{mn}$ and a similar relation exists for the rows of $\underline{P}$ [41], then $\underline{P}^{-1} = \underline{P}^H$ which implies $\underline{P}$ is a unitary matrix. Consequently, the similarity transformation $\underline{\Lambda} = \underline{P}^{-1}\,\underline{T}\,\underline{P}$ that diagonalizes $\underline{T}$ becomes

$$\underline{\Lambda} = \underline{P}^H\,\underline{T}\,\underline{P}\ , \qquad \text{(A.1)}$$

which is called a unitary transformation. Hermitian matrices can be easily decomposed into their spectral representations by using unitary transformations in the following way,

$$\underline{T} = \underline{P}\ \underline{\Lambda}\ \underline{P}^H = \sum_{m=1}^{N} \lambda_m \underline{p}_m \underline{p}_m^H \quad . \tag{A.2}$$

When the vector space is real and thus $\underline{T}$ is real symmetric, the complex conjugate transpose is simply the transpose. Results analogous to the hermitian case exist for the general case where $\underline{T}$ does not have N linearly independent eigenvectors, but the theory is considerably more involved [42].

A projection operator is represented by a <u>projection matrix</u> $\underline{T}$ which is hermitian and has the idempotent property $\underline{T}^2 = \underline{T}$. It can be shown [43] that all eigenvalues are either zero or unity, and that the eigenvectors of the unity eigenvalues span the subspace into which the operator projects. Thus any projection operator is similar to the block diagonal matrix $\mathrm{diag}(\underline{I}|\underline{0}) = \mathrm{diag}(1,1,\ldots 1,\ 0,\ldots 0)$ where the square matrices $\underline{I}$ and $\underline{0}$ are of orders equal to the number of unity and zero eigenvalues, respectively.

A <u>quadratic form</u> is a real-valued function of N real variables $\{x_m\}$ defined by

$$F(x_1,\ x_2,\ldots x_N) = \sum_{m=1}^{N} \sum_{n=1}^{N} r_{mn}\ x_m\ x_n\ , \tag{A.3}$$

which can also be written as

$$\sum_{m=1}^{N} r_{mn}\, x_m^{\,2} + \sum_{m=1}^{N} \sum_{n<m} (r_{mn} + r_{nm})\, x_m\, x_n .$$

The value of this last expression remains unchanged if the substitution $q_{mn} = 1/2 \cdot (r_{mn} + r_{nm})$ is made. By defining the real symmetric matrix $\underline{Q} \triangleq [q_{mn}]$ and letting $\underline{x} = (x_1, x_2, \ldots x_N)^T$, (A.3) can be written in matrix form as

$$F(\underline{x}) = \underline{x}^T \underline{Q}\, \underline{x} = \langle \underline{x}, \underline{Q}\, \underline{x} \rangle .$$

In the more general complex case when $\underline{Q}$ is hermitian the function is still a quadratic form since then

$$F(\underline{x}) = \sum_{m=1}^{N} q_{mm}\, x_m^{\,2} + \sum_{m=1}^{N} \sum_{n<m} (q_{mn} + q_{mn}^{\,*})\, x_m\, x_n ,$$

which is real due to $q_{mn} + q_{mn}^{\,*} = 2\, \mathrm{Re}(q_{mn})$.

A real symmetric matrix $\underline{Q}$ is positive semidefinite if, for any choice of $\underline{x}$, the quadratic form $F(\underline{x}) = \langle \underline{x}, \underline{Q}\, \underline{x} \rangle$ is nonnegative. If $F(\underline{x}) = 0$ only when $\underline{x} = \underline{0}$ then $\underline{Q}$ is positive definite. The eigenvalues of $\underline{Q}$ are related to the definiteness in the following way; if $\underline{Q}$ is positive semidefinite all eigenvalues are nonnegative, while all are strictly positive when $\underline{Q}$ is positive definite. If $\underline{Q}$ is positive definite it is invertible.

The final type of special matrix to be considered in this section is the Hadamard matrix, which has orthogonal rows and columns, and is binary in the sense that all

elements are either 1 or -1. Except for the 2 x 2 case

$$\underline{H}_0 \triangleq \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} ,$$

Hadamard matrices may exist only for orders which are integer multiples of four, and have been constructed for all such orders up to 100 [44]. For those orders which are also powers of two, the construction is done recursively by starting with $\underline{H}_0$ and using the iterative relation

$$\underline{H}_{k+1} \triangleq \left[\begin{array}{c|c} \underline{H}_k & \underline{H}_k \\ \hline \underline{H}_k & -\underline{H}_k \end{array}\right] .$$

The construction of other orders involves the determination of quadratic residues [45] and is thus somewhat more involved in theory but easily implemented using a digital computer as discussed in Section C.2.

A.3 THE PSEUDOINVERSE MATRIX

Before defining the pseudoinverse matrix a number of preliminary notions are necessary. Let the M x N matrix $\underline{X}$ represent a linear map from a vector space U to a normed vector space V of dimensions N and M, respectively. Assume that the field is the set of real numbers and that $M \geq N$. Then all eigenvalues of the real symmetric matrices $\underline{X}^T\underline{X}$ and $\underline{X}\underline{X}^T$ are real and nonnegative. The real property follows

from symmetry; to show that $\underline{X}^T\underline{X}$ has nonnegative eigenvalues, let $\underline{v} = \underline{X}\,\underline{u}$ for some arbitrary vector $\underline{u}$ in U. Then

$$0 \leq ||\underline{v}||^2 = \underline{v}^T\,\underline{v} = \underline{u}^T\,\underline{X}^T\underline{X}\,\underline{u} \quad ,$$

so $\underline{X}^T\underline{X}$ is positive semidefinite from which the desired result follows. An analogous procedure holds for $\underline{X}\underline{X}^T$. If the rank of $\underline{X}$ is $R \leq N$, then it can be shown [46] that both $\underline{X}^T\underline{X}$ and $\underline{X}\underline{X}^T$ have exactly R identical nonzero eigenvalues, say $\{\mu_1^2, \mu_2^2, \ldots \mu_R^2\}$, with all other eigenvalues zero. Furthermore, from symmetry it follows that $\underline{X}^T\underline{X}$ and $\underline{X}\underline{X}^T$ each have a set of orthonormal eigenvectors which will be denoted by the sets $\{\underline{\xi}_1, \underline{\xi}_2, \ldots \underline{\xi}_N\}$ and $\{\underline{\psi}_1, \underline{\psi}_2, \ldots \underline{\psi}_M\}$, respectively. These sets are ordered so that $\underline{\xi}_m$ and $\underline{\psi}_m$ correspond to μ_m^2 for $m = 1, 2, \ldots R$. The rectangular matrix $\underline{X}$ can now be expressed [47] in the generalized spectral representation

$$\underline{X} = \sum_{k=1}^{R} \mu_k\, \underline{\psi}_k\, \underline{\xi}_k^T \quad . \tag{A.4}$$

When $\underline{X}$ is real symmetric, hence square, then (A.4) reduces to (A.2). This can be shown by noting that $\underline{X}\underline{X}^T = \underline{X}^T\underline{X} = \underline{X}^2$ and, from the Frobenius Theorem the nonzero eigenvalues of $\underline{X}$ are $\{\mu_1, \mu_2, \ldots \mu_R\}$. If $\underline{p}$ is a normalized eigenvector of $\underline{X}$ corresponding to μ, it is also an eigenvector of $\underline{X}^2$ corresponding to μ^2. To show this note that $(\underline{X} - \mu\,\underline{I})\,\underline{p} = \underline{0}$ implies that $(\underline{X}^2 - \mu^2\underline{I})\underline{p} = \underline{X}\,(\underline{X}\,\underline{p}) - \mu\,\underline{X}\,\underline{p} = \underline{X}\,[(\underline{X} - \mu\,\underline{I})\,\underline{p}] = \underline{0}$. Consequently, for this case $\{\underline{p}_m\}$, $\{\underline{\xi}_m\}$, and $\{\underline{\psi}_m\}$ are

all equal and the desired result follows.

The N x N pseudoinverse matrix $\underline{X}^{-I}$ of $\underline{X}$ is now defined by

$$\underline{X}^{-I} \triangleq \sum_{k=1}^{R} \mu_k^{-1} \underline{\xi}_k \underline{\psi}_k^T . \qquad (A.5)$$

One important application of this definition can be seen by considering the matrix equation $\underline{X}\,\underline{u} = \underline{v}$. If $\underline{v}$ belongs to the range space of $\underline{X}$ then an exact solution exists, but is unique only when $\underline{X}$ is nonsingular in which case it is given by the usual inverse $\underline{X}^{-1}$ as $\underline{u} = \underline{X}^{-1}\,\underline{v}$. Otherwise there are more linearly independent equations than unknowns and an exact solution does not exist. In either case the pseudoinverse solution $\hat{\underline{u}} = \underline{X}^{-I}\,\underline{v}$ is the unique best solution in the sense that, for any other possible solution $\tilde{\underline{u}}$, it can be shown [48] that $\| \underline{X}\,\hat{\underline{u}} - \underline{v} \| \leq \| \underline{X}\,\tilde{\underline{u}} - \underline{v} \|$. Finally, there are two other properties of the pseudoinverse that will prove useful, both of which are easily proven from the basic equations (A.4) and (A.5). The properties are $(\underline{X}\underline{X}^{-I})^T = \underline{X}\underline{X}^{-I}$ and $\underline{X}\underline{X}^{-I}\,\underline{X} = \underline{X}$. Together they imply that

$$(\underline{X}\underline{X}^{-I})^T (\underline{X}\underline{X}^{-I}) = \underline{X}\underline{X}^{-I}. \qquad (A.6)$$

A.4 THE STANDARD Z-TRANSFORM

A thorough treatment of the z-transform is found in Jury [49], and its use in the design of digital filters is discussed by Gold and Rader [50]. The z-transform is a linear map from a vector space of sequences, $\underline{f} = \{f_m\}$, into a vector space of functions $F(z)$ of the complex variable z. It is defined by

$$F(z) = \sum_{k=0}^{\infty} f_k \, z^{-k} , \qquad (A.7)$$

which is a power series in z^{-1}, thus z^{-1} is taken within the radius of convergence of the power series. Multiplying $F(z)$ by z^{-1} has the effect of shifting the corresponding sequence $\{f_m\}$ by one index as is seen from

$$z^{-1} F(z) = \sum_{k=0}^{\infty} f_k \, z^{-(k+1)} = \sum_{n=1}^{\infty} f_{n-1} \, z^{-n} .$$

Consequently, z^{-1} is known as the unit delay operator. The inverse z-transform can be obtained by multiplying both sides of (A.7) by z^{m-1}, integrating around a closed contour that encloses the origin of the z-plane and applying the Cauchy integral theorem to obtain

$$\frac{1}{2\pi j} \oint F(z) \, z^{m-1} dz = \sum_{k=0}^{\infty} f_k \, \frac{1}{2\pi j} \oint z^{-(k-m+1)} dz = f_m .$$

When the sequence $\{f_m\}$ represents a stable physical quantity, for example a sampled voltage waveform, then it must not diverge. If F(z) is rational and contains terms of the form $z/(z-a)$, then stability requires that $|a| < 1$, or equivalently, that all poles of F(z) lie within the unit circle of the z-plane. Since this result can be extended to all stable sequences, the unit circle is the smallest contour that always encloses all singularities of F(z); consequently, the unit circle will always be taken as the path of integration in the z-plane.

There are two important properties which will be used extensively herein; namely, the multiplication relation and the Complex Convolution Theorem, known also as Parseval's Theorem. Multiplication in the z-transform domain is given by

$$H(z) = F(z)\ G(z) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} f_n g_m\ z^{-(m+n)}$$

$$= \sum_{k=0}^{\infty} \sum_{n=0}^{k} f_n g_{k-n}\ z^{-k},$$

so that from (A.7) it follows that

$$h_k = \sum_{n=0}^{k} f_n g_{k-n}\ . \qquad \text{(A.8)}$$

This last expression is the discrete convolution of the

sequences $\{f_m\}$ and $\{g_m\}$. The Complex Convolution Theorem can be obtained by considering the sequence $\{d_m\}$ obtained as term-by-term products of the two sequences $\{e_m\}$ and $\{f_m\}$, i.e., $d_k \triangleq e_k f_k$. The z-transform of $\{d_m\}$ is

$$D(z) = \sum_{k=0}^{\infty} e_k f_k z^{-k} = \sum_{k=0}^{\infty} e_k z^{-k} \frac{1}{2\pi j} \oint F(v)\, v^{k-1} dv$$

$$= \frac{1}{2\pi j} \oint F(v) \left[\sum_{k=0}^{\infty} e_k (z/v)^{-k} \right] v^{-1} dv$$

$$= \frac{1}{2\pi j} \oint F(v)\, E(z/v) v^{-1} dv, \qquad (A.9)$$

which is known as the <u>complex convolution</u> of $E(z)$ and $F(z)$. A particularly useful special case is obtained by first letting $z = 1$ to obtain

$$\sum_{k=0}^{\infty} e_k f_k = \frac{1}{2\pi j} \oint F(v)\, E(1/v) v^{-1} dv,$$

then setting $f_k = e_k$ for each k, and finally replacing the dummy variable v with z. The result is

$$\sum_{k=0}^{\infty} e_k^2 = \frac{1}{2\pi j} \oint E(z)\, E(z^{-1}) z^{-1}\, dz. \qquad (A.10)$$

This equation reduces to the classical form of Parseval's

Identity when the path of integration is the unit circle. This corresponds to letting $z=e^{j\omega}$ which gives

$$\sum_{k=0}^{\infty} e_k^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} E(j\omega)E(-j\omega)\,d\omega \quad ,$$

where $E(j\omega)$ is the Fourier transform of the continuous function $e(t)$ from which $\{e_m\}$ is obtained by sampling at the normalized rate of 1 sample per second. Equivalently ω can be considered as an angle defined by $\omega = \Omega T$, where Ω is the actual radian frequency and $1/T$ is the sampling frequency.

APPENDIX B

SUPPLEMENTARY RESULTS FOR CHAPTERS 2 AND 3

A number of statements from Chapters 2 and 3 are proven in this appendix. Although all of the proofs are important, inclusion in the chapters would have distracted from the chapter continuity. Most results are given in the "statement-proof" form, and the section from which the statement was taken is found in parenthesis after the proof number, e.g., "B.1 (2.2)" indicates that the first proof in Appendix B is of a statement in Section 2.2.

B.1 (2.2) The matrix $\underline{A}$ defined by

$$a_{mn} = \frac{1}{2\pi j} \oint X(z)\, F_m(z)\, F_n(z^{-1})\, X(z^{-1})\, z^{-1}\, dz$$

is hermitian and positive semidefinite.

Proof: By letting $z = e^{j\omega}$ it follows that

$$a_{mn} = \frac{1}{2\pi j} \int_{-\pi}^{\pi} X(j\omega)X(-j\omega)F_m(j\omega)F_n(-j\omega)\, j\, d\omega$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(j\omega)|^2\, F_m(j\omega)F_n^*(j\omega)\, d\omega = a_{nm}^*,$$

which implies that $\underline{A}$ is hermitian. To show positive semi-definiteness let $\underline{c}$ be an arbitrary M-vector, then

$$\underline{c}^T \underline{A} \underline{c} = \frac{1}{2\pi j} \oint \underline{c}^T \underline{V}(z) \underline{V}^T(z^{-1}) \underline{c} \, z^{-1} \, dz.$$

Since $Y(z) = \underline{c}^T \underline{V}(z)$, it follows that

$$\underline{c}^T \underline{A} \underline{c} = \frac{1}{2\pi j} \oint Y(z) \, Y(z^{-1}) \, z^{-1} \, dz$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} | Y(j\omega) |^2 \, d\omega \geq 0 .$$

Q.E.D.

B.2 (2.3) The matrix $\tilde{\underline{A}}$ defined by taking $F_m(z) = z^{-m}$ in (2.5) is positive definite.

Proof: Referring to Fig. 2.2, the filter output can be written as

$$Y(z) = \sum_{m=0}^{M-1} c_m F_m(z) X(z) = \left[\sum_{m=0}^{M-1} c_m z^{-m} \right] X(z)$$

$$= C(z) X(z),$$

where $C(z)$ is defined to be the term in brackets. Using this notation the quadratic form $\underline{c}^T \tilde{\underline{A}} \underline{c}$ can be written

$$\underline{c}^T \tilde{\underline{A}} \underline{c} = \frac{1}{2\pi j} \oint Y(z) \, Y(z^{-1}) \, z^{-1} \, dz$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} | C(j\omega) |^2 \, | X(j\omega) |^2 \, d\omega.$$

Now $C(j\omega)$ is a linear combination of M complex exponentials, and hence must always be nonzero on any finite interval unless $\underline{c} = \underline{0}$; furthermore, the input signal is converted to digital form by an A/D converter operating at a sampling rate which is at least twice the highest signal frequency in accordance with the uniform sampling theorem. Consequently, the spectrum $X(j\omega)$ must be nonzero on some interval of the primary region $[-\pi,\pi]$, and therefore, $\underline{c}^T \underline{A} \underline{c} > 0$ for $\underline{c} \neq \underline{0}$ and $\tilde{\underline{A}}$ is positive definite. Q.E.D.

B.3 (2.3) One-step convergence can be obtained for the transversal equalizer using the more general algorithm

$$\underline{c}[k+1] = \underline{c}[k] - 1/2 \ \underline{\Gamma}[k] \ \nabla J(\underline{c}[k])$$

where $\underline{\Gamma}[k] \triangleq \text{diag}(\gamma_0[k], \gamma_1[k], \ldots \gamma_{M-1}[k])$.

Proof: A derivation analogous to that of (2.14) for this new algorithm gives

$$\underline{e}[k] = \prod_{i=0}^{k-1} (\underline{I} - \underline{\Gamma}[i] \tilde{\underline{A}}) \ \underline{e}[0]$$

Letting the definitions of $\tilde{\underline{\Lambda}}$ and $\tilde{\underline{Q}}$ remain unchanged, the result corresponding to (2.15) is

$$J_{cor}[k] = \sum_{m=0}^{M-1} (\underline{e}^T[0] \ \tilde{\underline{q}}_m)^2 \ \tilde{\lambda}_m \prod_{i=0}^{k-1} (1 - \gamma_m[i] \ \tilde{\lambda}_m)^2,$$

which reduces to zero on the first iteration upon adjusting the diagonal components of $\underline{\Gamma}[0]$ to satisfy $\gamma_m[0] = \tilde{\lambda}_m^{-1}$ for m = 0,1,...M-1.

Q.E.D.

B.4 (2.4) For equation (2.20), the vector $\underline{b}$ belongs to the range space $R(\underline{A})$ of $\underline{A}$.

Proof: Since $\underline{b} = \underline{F}^T \tilde{\underline{b}}$ and $\underline{A} = \underline{F}^T \tilde{\underline{A}} \underline{F}$, it is sufficient to show $R(\underline{A}) = R(\underline{F}^T)$. Since $\tilde{\underline{A}}$ is nonsingular and real symmetric it has a square root $\tilde{\underline{A}}^{1/2}$ which is also nonsingular and real symmetric; consequently, $\underline{A}$ can be written as $\underline{A} = (\tilde{\underline{A}}^{1/2}\underline{F})^T (\tilde{\underline{A}}^{1/2}\underline{F}) \triangleq \underline{S}^T \underline{S}$. Also, $\underline{b}$ can be written $\underline{b} = \underline{F}^T(\tilde{\underline{A}}^{1/2} \tilde{\underline{A}}^{-1/2}) \tilde{\underline{b}} = \underline{S}^T (\tilde{\underline{A}}^{-1/2} \tilde{\underline{b}})$. Upon observing that $\underline{S}^T$ is the adjoint of $\underline{S}$ it follows from A.1 that $R(\underline{S}^T) = R(\underline{S}^T\underline{S}) = R(\underline{A})$. However, $\underline{b}$ is obtained by applying $\underline{S}^T = \underline{F}^T \tilde{\underline{A}}^{1/2}$ to the arbitrary vector $\tilde{\underline{A}}^{-1/2} \tilde{\underline{b}}$ and since $\tilde{\underline{A}}^{1/2}$ is one-to-one, then $R(\underline{S}^T) = R(\underline{F}^T)$. Consequently, $R(\underline{F}^T) = R(\underline{S}^T) = R(\underline{A})$.

Q.E.D.

B.5 (2.5) The map $\underline{P} = \underline{X}\underline{X}^{-I}$ defined by (2.33) is a projection matrix, i.e., it is both self-adjoint and idempotent.

Proof: First, $\underline{P}$ is hermitian since it is real and

$$\underline{P}^T = \left[\sum_{m=0}^{N-1} \underline{\psi}_m \underline{\psi}_m^T\right]^T = \sum_{m=0}^{N-1} \underline{\psi}_m \underline{\psi}_m^T = \underline{P} \quad .$$

Furthermore, since

$$\underline{P}^2 = \left(\sum_{m=0}^{N-1} \underline{\psi}_m \, \underline{\psi}_m^T \right) \left(\sum_{n=0}^{N-1} \underline{\psi}_n \, \underline{\psi}_n^T \right)$$

$$= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \underline{\psi}_m \, \delta_{mn} \, \underline{\psi}_n^T = \underline{P} \ ,$$

then $\underline{P}$ is also indempotent, hence $\underline{P}$ represents a projection operator. Q.E.D.

B.6 (2.5) The matrix $\underline{\Psi}^T \, \underline{P} \, \underline{\Psi} = \text{diag}(\underline{I}_N \mid \underline{0}_{M-N})$.

Proof: The result follows directly from the fact that $\{\underline{\psi}_m\}$ is an orthonormal set:

$$\underline{\Psi}^T \, \underline{P} \, \underline{\Psi} = \begin{bmatrix} \underline{\psi}_0^T \\ \underline{\psi}_1^T \\ \vdots \\ \underline{\psi}_{M-1}^T \end{bmatrix} \sum_{m=0}^{N-1} \underline{\psi}_m \, \underline{\psi}_m^T \left[\underline{\psi}_0, \, \underline{\psi}_1, \cdots \underline{\psi}_{M-1} \right]$$

$$= \left[\begin{array}{c|c} \underline{I}_N & \underline{0} \\ \hline \underline{0} & \underline{0}_{M-N} \end{array} \right] = \text{diag}(\underline{I}_N \mid \underline{0}_{M-N}) .$$

Q.E.D.

B.7 (3.1) If a priori statistical information is available for evaluation of the gradient (3.4), then the tap-gain adjustment algorithm (3.2) reduces to other well-established algorithms.

In the first case it is assumed that the noise statistics are unknown, the transmitted signal and noise are uncorrelated, and that statistical information is known about the transmitted sequence $\{d[k]\}$. Under the decision-directed hypothesis that $\hat{d}[k] = d[k]$ for all k, the gradient (3.4) becomes

$$\frac{1}{2} \nabla \bar{J}(\underline{c}[k]) = E\{\underline{v}[k]\ \underline{v}^T[k]\}\ \underline{c}[k] - \underline{r}_{dv} \quad ,$$

where $\underline{c}[k]$ is assumed independent of $\underline{v}[k]$ and

$$\underline{r}_{dv} = \underline{r}_{dv}(n) \triangleq E\{\underline{v}[k]\ \hat{d}[k]\} = \underline{F}^T\ E\{\underline{x}[k]\ d[k]\}.$$

The result, known as the Griffiths Algorithm, was considered by Griffiths [51] in connection with antenna-array signal processing, and some convergence results were developed.

For the second case it is assumed that complete statistical information is available a priori, so that (3.4) becomes

$$\frac{1}{2} \nabla \bar{J}(\underline{c}[k]) = \underline{R}_{vv}\ \underline{c}[k] - \underline{r}_{dv},$$

with $\underline{r}_{dv}$ defined as before and the M x M correlation matrix $\underline{R}_{vv}$ given by $\underline{R}_{vv} = \underline{R}_{vv}(0) \triangleq E\{\underline{v}[k]\ \underline{v}^T[k]\}$. The result is the well-known steepest descent algorithm

$$\underline{c}[k+1] = \underline{c}[k] - \gamma\ (\underline{R}_{vv}\ \underline{c}[k] - \underline{r}_{dv}).$$

B.8 (3.2) Under the stated hypothesis the matrix $\underline{R}_{xx}$ is positive definite.

Proof: Since $x[k] = s[k] + n[k]$ for all k, then the signal and noise vectors $\underline{s}[k]$ and $\underline{n}[k]$, respectively, corresponding to $\underline{x}[k]$ are defined component-wise by $\underline{x}[k] = \underline{s}[k] + \underline{n}[k]$. It follows from the independence of signal and noise that $\underline{R}_{xx} = E\{\underline{s}[k]\ \underline{s}^T[k]\} + E\{\underline{n}[k]\ \underline{n}^T[k]\} \triangleq \underline{R}_{ss} + \underline{R}_{nn}$. Now $\underline{R}_{ss}$ is a correlation matrix, hence positive semidefinite; also $\underline{R}_{nn}$ can be diagonalized by a unitary matrix $\underline{Q}$ to give $\underline{Q}^T\ \underline{R}_{xx}\ \underline{Q} = \underline{\Lambda}_{nn}$. However, upon defining $\underline{m} \triangleq \underline{Q}^T\underline{n}$ a new representation of the noise results with crosscorrelation matrix $E\{\underline{Q}^T\underline{n}[k]\underline{n}^T[k]\underline{Q}\} = \underline{Q}^T\underline{R}_{nn}\ \underline{Q} = \underline{\Lambda}_{nn}$, i.e., the crosscorrelation matrix of $\underline{m}[k]$ is diagonal. The elements are all variances, thus are all positive, and it follows that $\underline{R}_{nn}$ and $\underline{R}_{xx}$ are positive definite.

Q.E.D.

APPENDIX C

COMPUTER SIMULATION

This appendix contains a detailed description of the digital computer simulation written in FORTRAN to study the equalizer design which has been developed. The simulation consists of a main routine, called MAIN, and a number of subroutines, which together simulate a transmitter, digital communication channel, and the equalizer. In addition to using subroutine GRAPH for producing graphical output of the equalizer performance, MAIN uses two other subroutines, namely FLTSET and HADGEN, to compute the initial tap-gain settings in accordance with (2.37). Each of these in turn calls two or more subroutines as shown in Fig. C.1. MAIN is described in C.1 while the remaining subroutines are covered in C.2. Two numerical methods for finding eigenvalues and eigenvectors of real symmetric matrices are described in C.3. One is used by the IBM library subroutine EIGEN [52] and the other by the UNIVAC library subroutines TRIDMX, EIGVAL, and EIGVEC [53]. It is interesting to note here that, since a modern digital computer is actually a very sophiscated digital filter in itself, then the simulation of the equalizer is quite realistic, although not very efficient from a hardware standpoint.

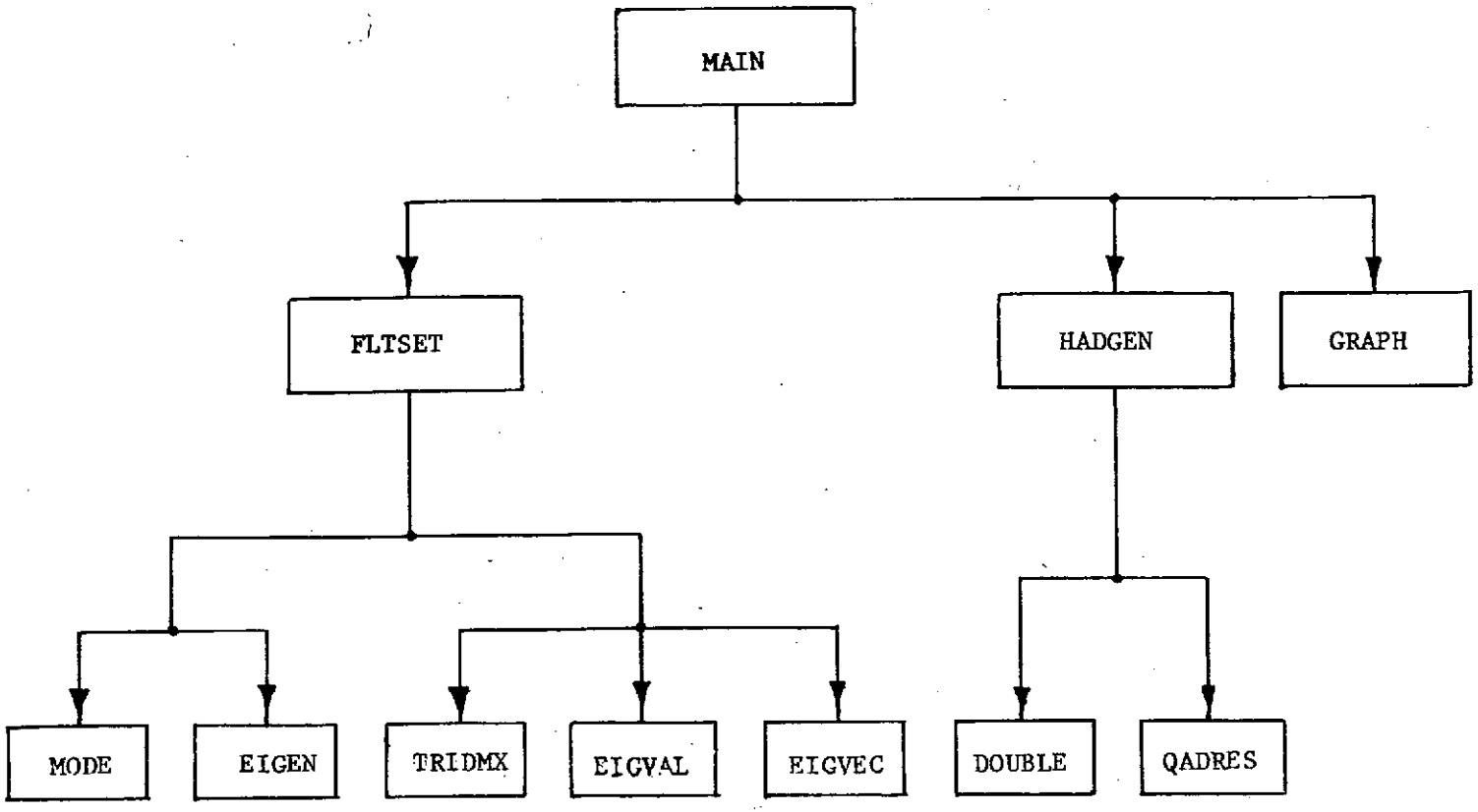

Fig. C.1. Hierarchy of Subroutines

C.1 PROGRAM MAIN

The program MAIN can be subdivided into two basic parts. The first simulates the transmission of an initializing pulse through the channel and performs the equalizer initialization based on the theory of Chapter 2. The second part generates the transmitter message sequence and uses the initialized equalizer to track the channel. A flowchart for MAIN is shown in Fig. C.2, and the program listing in Fig. C.3. For reasons of continuity a few parameters, usually pertaining to small blocks of code, are set locally as they occur rather than at the beginning of the program. The listing of key parameters found in Table C.1 will be helpful in identifying and using these variables.

A concise description of the program is found in block 0. The program is designed to repeat until no further data cards are found, at which time it terminates normally. After reading the simulation parameters and initializing variables in blocks 0-30, the channel output sequence (X) is formed in blocks 40 and 50 by passing an initializing pulse through the noisy channel (CHNL, G, CNOISE). Block 60 calls the subroutine FLTSET which returns in common the eigenvalue and eigenvector information (FSET) for computing (2.37). In block 70 an M x M Hadamard matrix (H) is formed by subroutine HADGEN, and the matrix (F), whose

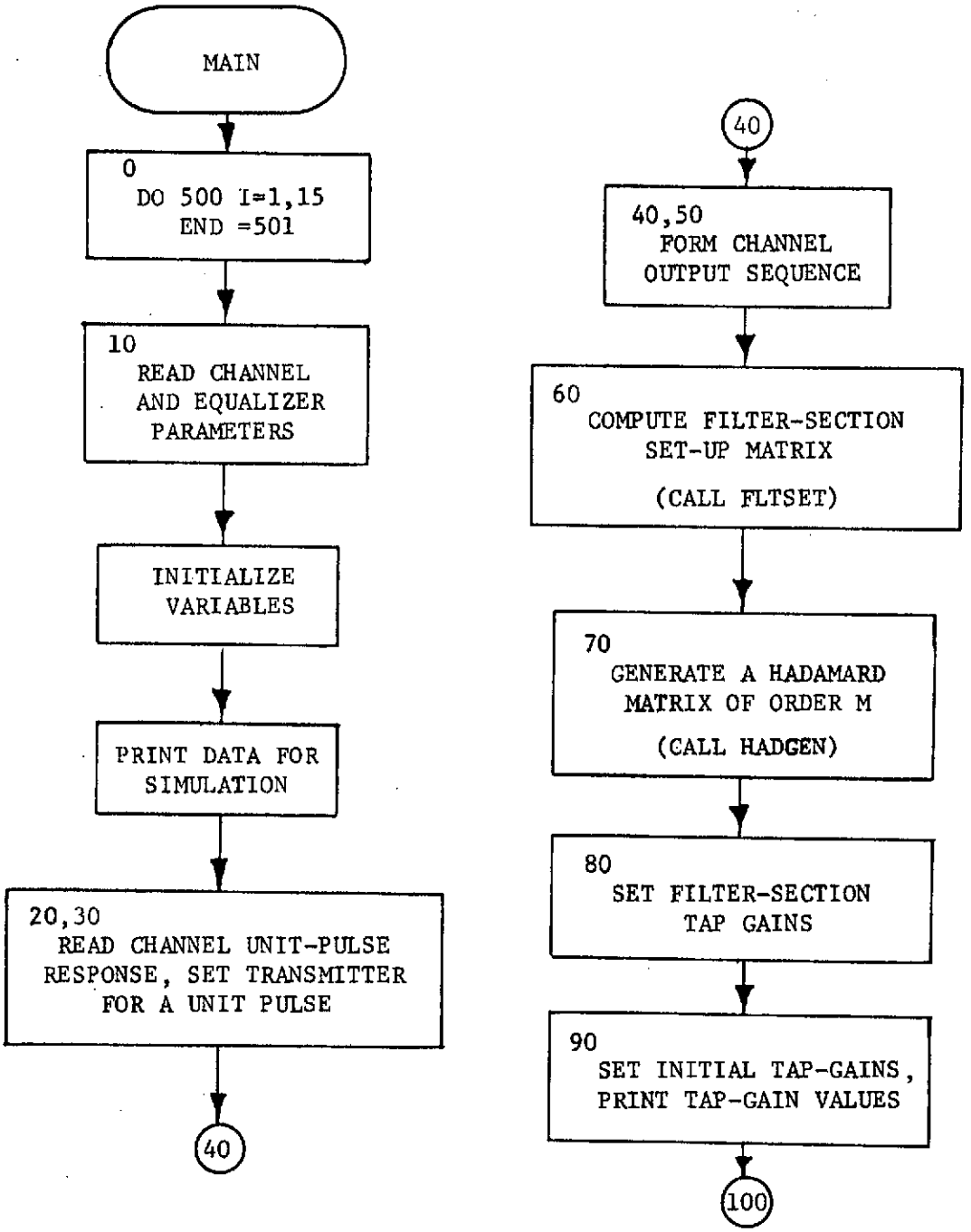

Fig. C.2. Simulation Program MAIN

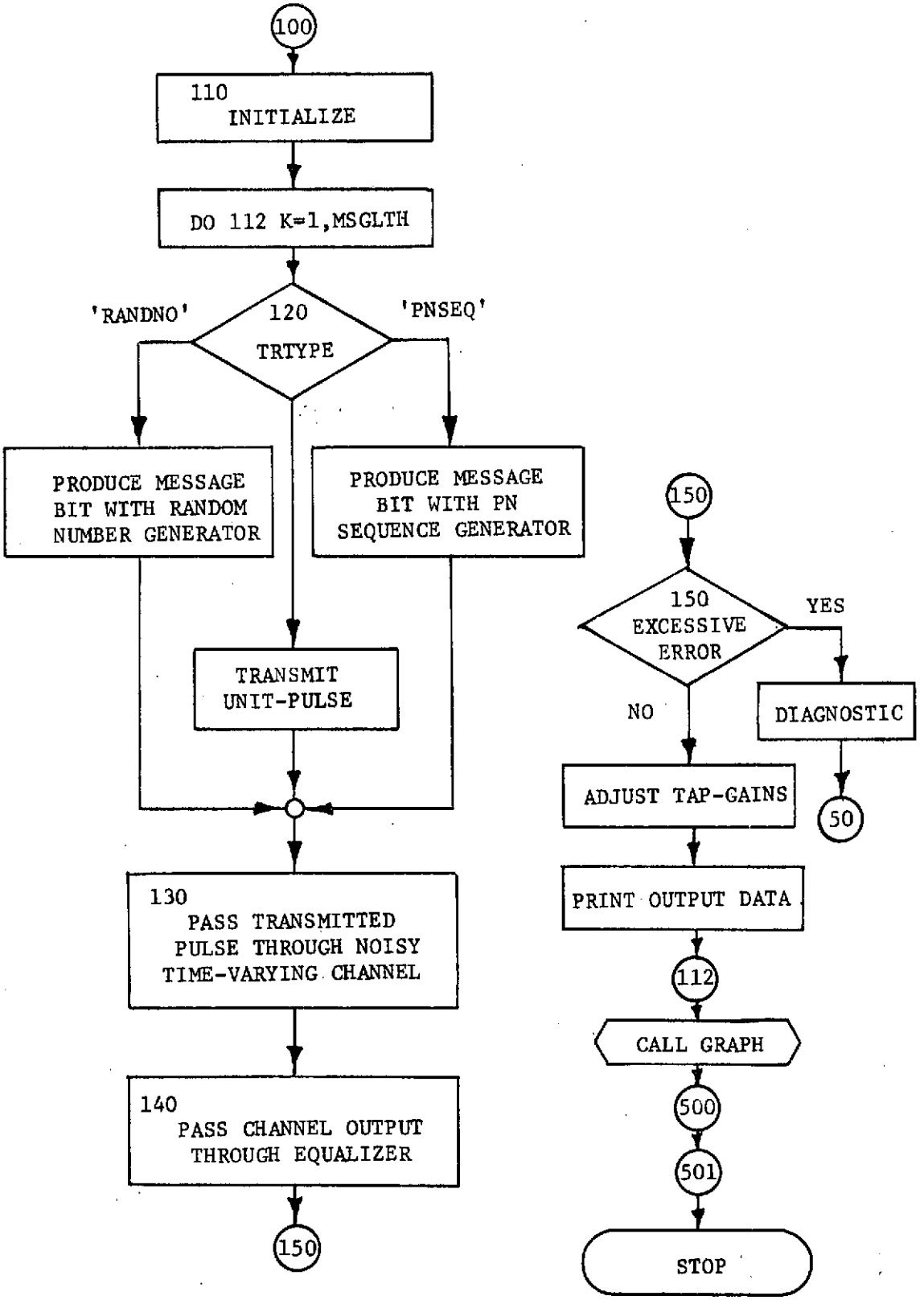

Fig. C.2. (Continued)

```
      DIMENSION D(32),X(32),G(32),CHNL(32),IREG(10),C(32),
     1 W(32),R(32),F(32,32),DELC(32),TRANS(32),PLOTY(64)
      INTEGER DELAY,DMAX,TRTYPE,TOTLER
      COMMON H(32,32),FSET(32,32)
      EQUIVALENCE (DELC(1),CHNL(1),H(1,1)),(TRANS(1),H(1,2))
     1 ,(PLOTY(1),H(1,3))
C
C000  THIS PROGRAM SIMULATES AN ADAPTIVE DIGITAL EQUALIZER
C     AND ASSOCIATED BASEBAND CHANNEL.  THE INITIALIZATION
C     IS PERFORMED IN BLOCKS 10-90, AND ON-LINE OPERATION
C     IS SIMULATED BY BLOCKS 100-150.
C
C
C     THE DATA CARD FORMATS ARE AS FOLLOWS
CARD1 M(I5), N(I5), DELAY(I5), MSGLTH(I5), TRTYPE(A6)
CARD2 CMEAN(F10.5), CSTDEV(F10.5), CVFCTR(F10.5), NINIT(I5)
CARD3 CHANNEL IMPULSE RESPONSE LENGTH NG(I5)
CARD4 CHANNEL IMPULSE RESPONSE G(8F10.5)
C
C     REPEAT ENTIRE SIMULATION UNTIL DATA RUNS OUT
      DO 500 II=1,15
      READ (5, 1,END=501) M,N,DELAY,MSGLTH,TRTYPE,
     1 CMEAN,CSTDEV,CVFCTR,NINIT
 1    FORMAT (415,4X,A6/3F10.5,I5)
C
C     INITIALIZE VARIABLES AND ARRAYS
      DO 10 I=1,32
      CHNL(I)=0
      C(I)=0
      D(I)=0
      G(I)=0
      X(I)=0
      DO 10 J=1,32
      F(I,J)=0
      H(I,J)=0
 10   CONTINUE
C     SET THE NOISE GENERATOR PARAMETERS
      IA=127329
      IB=566387
      IC=2**20
      REALC=IC
      CVAR=CSTDEV*CSTDEV
C     PROGRAMMING NOTE: CSTDEV MUST NOT BE ZERO
      SNR= 20*ALOG10(1/CSTDEV)
C     PRINT THE INITIAL EQUALIZER PARAMETERS
      WRITE (6,12) M,N,MSGLTH,TRTYPE,CMEAN,CVAR,SNR,DELAY,
```

Fig. C.3. Listing of Program MAIN

```
     1 CVFCTR,NINIT
  12  FORMAT (1H1////10X, 'SYSTEM PARAMETERS'/
     1 10X,'NUMBER OF FILTER SECTIONS (M) =',I5/
     2 10X,'LENGTH OF FILTER SECTIONS (N) =',I5/
     3 10X,'LENGTH OF TRANSMITTED MESSAGE (MSGLTH) =',I7/
     4 10X,'TYPE OF TRANSMITTER SIMULATED (NTRTYP) =',A6/
     5 10X,'NOISE MEAN =',F10.5/10X,'NOISE VARIANCE =',F10.5/
     6 10X,'SIGNAL TO NOISE RATIO IN DB =',F10.5/
     7 10X,'TOTAL DELAY =', I5  /10X,'CONVERGENCE FACTOR =',
     8 F10.5/10X,'NUMBER OF INITIALIZING PULSES =',I5)
C
C020  READ IN THE CHANNEL DESCRIPTION
C
      READ (5,21) NG, (G(I),I=1,NG)
  21  FORMAT ( I5/(8F10.5))
      WRITE (6,22) NG, (I,G(I),I=1,NG)
  22  FORMAT (////10X,'CHANNEL RESPONSE G OF LENGTH',I5/
     1 (10X,'G(',I2,')=',F8.4))
C
C030  IF INITIALIZING SEQUENCES OTHER THAN THE UNIT-PULSE
C     ARE USED THEY SHOULD BE DEFINED IN THIS BLOCK,
C     OTHERWISE A UNIT-PULSE SEQUENCE IS GENERATED
C
      ND=1
      D(1)=1
C
C040  ADJUST THE INPUT SEQUENCE LENGTH
C
C        CHECK FOR EXCESSIVE CHANNEL-RESPONSE LENGTH
      L=ND+NG-1
      IF (L .LE. M-N+1) GO TO 41
      WRITE(6,42)
  42  FORMAT (1H1,10X,'***THE OUTPUT IS TOO LONG***')
      GO TO 500
C     EXTEND THE OUTPUT SEQUENCE LENGTH
  41  L=M-N+1
C
C050  FORM THE CHANNEL OUTPUT SEQUENCE X(Z)=D(Z)*G(Z)
C
  50  CONTINUE
      DO 52 I=1,NG
  52  CHNL(I)=0
      DO 51 K=1,L
      NGM=NG-1
      DO 53 I=1,NGM
  53  CHNL(NG-I+1)=CHNL(NG-I)
      CHNL(1)=D(K)
```

Fig. C.3. (Continued) Listing of Program MAIN

```
      X(K)=0
      DO 54 I=1,NG
 54   X(K)=X(K)+G(I)*CHNL(I)
C     GENERATE A GAUSSIAN RANDOM VARIABLE FOR CHANNEL NOISE
      RSUM=0
      DO 55 I=1,12
      IB=MOD(IB*(2**10+3),IC)
      REALB=IB
 55   RSUM=RSUM+REALB/REALC
      CNOISE=(RSUM-6)*CSTDEV/NINIT + CMEAN
C     ADD THE CHANNEL NOISE
      X(K)=X(K)+CNOISE
 51   CONTINUE
C     PRINT THE CHANNEL OUTPUT FOR THE UNIT-PULSE
      WRITE (6,56) L,(I,X(I),I=1,L)
 56   FORMAT (////10X,'CHANNEL OUTPUT SEQUENCE X OF LENGTH',
     1 I5/(10X'X(',I2,')=',F8.4))
C
C060  COMPUTE THE FILTER SETUP MATRIX FSET
C
C     SELECT TYPE OF EIGENVALUE COMPUTATION
      EIGTYP='JACOBI'
      CALL FLTSET (M,N,X,EIGTYP)
C
C070  GENERATE AN MXM HADAMARD MATRIX AND NORMALIZE
C
      CALL HADGEN(M)
      REALM=M
      SQM=SQRT(REALM)
      DO 70 I=1,M
      DO 70 J=1,M
      H(I,J)=H(I,J)/SQM
 70   CONTINUE
C
C080  SET THE FILTER SECTIONS
C
      DO 80 J=1,M
      DO 80 I=1,N
      F(I,J)=0
      DO 81 K=1,N
 81   F(I,J)=F(I,J)+FSET(I,K)*H(K,J)
 80   CONTINUE
C
C090  INITIALIZE THE FILTER AND SET THE TAP GAINS
C
      DO 91 I=1,M
      DELC(I)=0
```

Fig. C,3. (Continued) Listing of Program MAIN

```
 91   R(I)=0
      AMSE=0
      DO 92 K=1,M
C     LOAD AND SHIFT THE FILTER REGISTERS
      NM=N-1
      DO 93 I=1,NM
 93   R(N-I+1)=R(N-I)
      R(1)=X(K)
C     FORM THE FILTER OUTPUT Y
      Y=0
      DO 94 I=1,M
      W(I)=0
      DO 95 J=1,N
 95   W(I)=W(I)+R(J)*F(J,I)
 94   Y=Y+W(I)*C(I)
C     COMPUTE THE BIT ERROR AND THE TAP ADJUSTMENTS
      E=Y
      IF (K.GT.DELAY) E=Y-D(K-DELAY)
      AMSE=AMSE+(E*E-AMSE)/K
      DO 96 I=1,M
 96   DELC(I)=DELC(I)+E*W(I)
 92   CONTINUE
C     SET THE EQUALIZER TAPS
      DO 97 I=1,M
 97   C(I)=C(I)-DELC(I)
      WRITE (6,98) (I,C(I),I=1,M)
 98   FORMAT (////10X,'INITIAL TAP GAIN SETTINGS'/
     1 (10X,'C(',I2,')=',F8.4))
      WRITE (6,99) MSGLTH
 99   FORMAT (1H1,25X,'THE TRANSMITTED SEQUENCE IS',I4,
     1 ' BITS LONG'////6X,11HTRANS/DEC'N,9X,'NOISE',4X,
     2 'CHNL OUT',2X,'EQLR OUT',4X,'ERROR',3X,'AVE MSE',
     3 2X,'TOTAL MSE',4X,'C(1)---->'//)
C
C100  THE REMAINING PART OF THIS PROGRAM IS A LARGE LOOP
C     THAT ITERATES ONCE FOR EACH MESSAGE BIT
C
C110  SHIFT TO RUN MODE AND TRANSMIT THE MESSAGE
C
      DO 111 I=1,L
      TRANS(I)=0
 111  CHNL(I)=0
      DO 113 I=1,64
 113  PLOTY(I)=0
      AMSE=0
      TMSE=0
      TOTLER=0
```

Fig. C.3. (Continued) Listing of Program MAIN

```
      NCOUT=4
      LL=8
      LLM=LL-1
      DMAX=25
      ELIM=2.0
      NGM=NG-1
      NLINE=1
C
      DO 112 K=1,MSGLTH
C
C120  GENERATE A TRANSMITTER PULSE
C
C     SELECT THE INPUT SEQUENCE FROM EITHER A PN
C     SEQUENCE (PNSEQ) OR RANDOM NUMBER GENERATOR (RANDNO)
      IF (TRTYPE.EQ.'PNSEQ') GO TO 122
      IF (TRTYPE.EQ.'RANDNO') GO TO 121
C     ASSUME THAT A UNIT-PULSE IS DESIRED
        DD=0
        IF (K.EQ.1) DD=1
      GO TO 126
 121  CONTINUE
C     GENERATE A RANDOM NUMBER BETWEEN 0 AND 1 USING
C     THE POWER RESIDUE METHOD
      IA=MOD(IA*(2**10+3),IC)
      REALA=IA
      RANDIN=REALA/REALC
C     CONVERT TO BINARY USING A THRESHOLD OF 0.5
      DD=0
      IF(RANDIN .LT. 0.5) DD=1
      GO TO 126
C     GENERATE A 2**LL-1 BIT PN SEQUENCE FOR THE MESSAGE
 122  IF (K.NE.1) GO TO 123
      DO 125 I=1,LL
 125  IREG(I)=1
 123  DD=IREG(3)
      ITEMP=MOD(IREG(2)+IREG(3)+IREG(4)+IREG(8),2)
      DO 124 I=1,LLM
 124  IREG(LL-I+1)=IREG(LL-I)
      IREG(1)=ITEMP
 126  CONTINUE
C
C     STORE THE TRANSMITTED SEQUENCE FOR PRINTOUT LATER
      DO 127 I=1,DMAX
 127  TRANS(DMAX-I+1)=TRANS(DMAX-I)
      TRANS(1)=DD
C
C130  PASS THE TRANSMITTED PULSE THROUGH THE TIME-VARYING
```

Fig. C.3. (Continued) Listing of Program MAIN

```
C       CHANNEL
C
 130    CONTINUE
C       THE TIME-VARYING CHANNEL SHOULD BE CHANGED HERE
        DO 131 I=1,NGM
 131    CHNL(NG-I+1)=CHNL(NG-I)
        CHNL(1)=DD
        XX=0
        DO 132 I=1,NG
 132    XX=XX+G(I)*CHNL(I)
C
C       GENERATE A GAUSSIAN RANDOM VARIABLE WITH PRESET
C       MEAN AND STANDARD DEVIATION
        RSUM=0
        DO 133 I=1,12
        IB=MOD(IB*(2**10+3),IC)
        REALB=IB
 133    RSUM=RSUM+REALB/REALC
        CNOISE=(RSUM-6)*CSTDEV+CMEAN
C       ADD THE CHANNEL NOISE
        XX=XX+CNOISE
C
C140    FILTER THE CHANNEL OUTPUT PULSE XX
C
C       LOAD AND SHIFT THE FILTER REGISTERS
        DO 141 I=1,NM
 141    R(N-I+1)=R(N-I)
        R(1)=XX
C       FORM THE FILTER OUTPUT Y
        Y=0
        DO 142 I=1,M
        W(I)=0
        DO 143 J=1,N
 143    W(I)=W(I)+R(J)*F(J,I)
 142    Y=Y+W(I)*C(I)
C       SET UP THE EQUALIZER OUTPUT FOR PLOTTING
        I=20*( Y+0.5)+1.5
        IF(I.LT.1) I=1
        IF(I.GT.51) I=51
        PLOTY(I)=PLOTY(I)+1
C
C150    COMPUTE THE ON-LINE TAP-GAIN ADJUSTMENTS
C
C       ESTIMATE THE TRANSMITTED VALUE
        ESTD=0
        IF(Y.GE.0.5) ESTD=1
        E=Y-ESTD
```

Fig. C.3. (Continued) Listing of Program MAIN

```
      AMSE=AMSE+(E*E-AMSE)/K
      TMSE=TMSE+E*E
C     CHECK FOR DIVERGENCE
      IF (ABS(E) .LT. ELIM) GO TO 153
      WRITE (6,151) K
 151  FORMAT (////10X,'***THE ALGORITHM DIVERGES AT STEP',
     1 I5,'  ***')
      GO TO 500
C     ADJUST THE FILTER TAPS USING THE GRADIENT ESTIMATES
 153  DO 152 I=1,M
 152  C(I)=C(I)-E*W(I)*CVFCTR
C     PRINT EVERY NLINE-TH LINE FOR BREVITY
        IF (K.NE.K/NLINE*NLINE) GO TO 155
      WRITE (6,154) K,TRANS(1+DELAY),ESTD,CNOISE,XX,Y,E,
     1 AMSE, TMSE,(C(I),I=1,NCOUT)
 154  FORMAT (1H ,I6,F4.1,' /',F4.1,6X,4F10.5,2G10.4,' *',
     1 4F10.5/(84X,4F10.5))
 155  CONTINUE
      IF (ABS(ESTD-TRANS(1+DELAY)) .LT. 0.1) GO TO 157
      TOTLER=TOTLER+1
      WRITE (6,156)
 156  FORMAT (1H+,1 X,'*****')
 157  CONTINUE
C
 112  CONTINUE
C
C500  TERMINATING PROCEDURE
C
      WRITE (6,502) TOTLER,MSGLTH
 502  FORMAT (//26X,'THERE WERE',I4,' ERRORS IN THE',I4,
     1 ' BIT MESSAGE')
      CALL GRAPH (-0.5,0.05,PLOTY ,51)
 500  CONTINUE
 501  CONTINUE
      STOP
      END
```

Fig. C.3. (Continued) Listing of Program MAIN

TABLE C.1

LIST OF SIMULATION VARIABLES

Name	Type[†]	Description
AMSE	R	Average MSE per iteration
CHNL(32)	R	Shift-register for simulating the channel
CMEAN	R	Mean value of channel noise
CNOISE	R	Channel noise value
CSTDEV	R	Standard deviation of channel noise
CVAR	R	Variance of channel noise
CVFCTR	R	Convergence factor for the on-line algorithm
D(32)	R	Working array for the initializing sequence
DELAY	I	Delay of reference pulse in bits
DELC(32)	R	Tap-gain increments for initialization
DMAX	I	Maximum delay
E	R	Equalizer output error
ELIM	R	Equalizer output error threshold
ESTD	R	Decision on equalizer output
F(32,32)	R	Matrix whose columns are the filter-section tap-gains
FSET(32,32)	R	Matrix returned by FLTSET used for computing initial filter-section tap-gains
G(32)	R	Noise-free channel unit-pulse response
H(32,32)	R	Hadamard matrix returned by HADGEN

[†]R indicates a real variable and I indicates an integer.

TABLE C.1 (Continued)

Name	Type	Description
L	I	Channel output sequence length
M	I	Number of filter sections
MSGLTH	I	Number of bits in transmitted message
N	I	Length of shift register
NCOUT	I	Number of tap-gain values to be printed at each iteration of the on-line algorithm
NG	I	Length of noise-free channel unit-pulse response
NINIT	I	Number of isolated initializing pulses used to set the equalizer
NLINE	I	Number of output lines skipped before printing a line of output data
PLOTY(60)	R	Contains the equalizer output-value distribution for plotting by GRAPH
R(32)	R	Shift-register for equalizer filter-sections
SNR	R	Channel SNR on a per-bit basis
TMSE	R	Total (cumulative) MSE
TOTLER	I	Total number of errors in equalizer output
TRANS(32)	R	Array to store transmitted sequence for later printout
TRTYPE		Transmitter type, either 'PNSEQ' or 'RANDNO' for PN sequence or random number generator, respectively; otherwise a unit-pulse sequence is transmitted
W(32)	R	Filter-section output vector
X(32)	R	Channel output sequence
Y	R	Equalizer output

columns are the filter-section tap-gains, is then computed as the product of FSET and H. The equalizer is now initialized in block 80 by first computing the equalizer tap-gain corrections (DELC), and then adding the result to then existing tap-gains (C) in accordance with (2.2), and the initial tap-gains are printed in block 90.

The remaining part of MAIN, starting with a description in block 100, is essentially a large loop that iterates once for each transmitted pulse. All coding inside the loop has been done inline, i.e., there are no subroutines, in order to reduce the program execution time. After a number of variables are initialized in block 110, the loop begins. A transmitter pulse (DD) is generated in block 120 using either a pseudo-noise sequence (TRTYPE = 'PNSEQ') or a random binary number (TRTYPE = 'RANDNO') generated by quantizing random numbers that are uniformly distributed between 0 and 1. If no transmitter type is specified then a unit-pulse is transmitted. In block 130 the pulse is next passed through the channel which is corrupted by gaussian noise. Since time-variation in communication channels is rather arbitrary, this effect is simulated by inserting a user-supplied description immediately after statement 130. The transmitted pulse (XX) is then filtered by the equalizer in block 140 to produce the equalizer output (Y). The average MSE per bit (AMSE) and total

(cumulative) MSE (TMSE) are next calculated, and the bit error (E) is compared against the maximum acceptable error value (ELIM). If the limit is exceeded an error message is printed and the equalizer can either be reinitialized by returning to block 50, or the run can be terminated by a transfer to 500. Otherwise, at block 150 the equalizer tap-gains are adjusted in accordance with (3.6) and, when indicated by a local parameter (NLINE), a line of data concerning the iteration is printed before repeating the loop for the next transmitter bit. After the indicated number (MSGLTH) of message bits has been transmitted, the total number of bit errors is printed and GRAPH is called to plot the frequency of occurrence of each value of the equalizer output.

C.2 SUBROUTINES

The flow chart and listing for subroutine FLTSET are found in Figs. C.4 and C.5, respectively. It generates the convolution matrix (XMX) of (2.28) from the channel unit-pulse response vector (X) that is passed from MAIN. The N x N real symmetric matrix (SYMX) is then formed and the corresponding eigenvalues and eigenvectors determined by one of the two methods discussed in C.3. If the Jacobi method is used (EIGTYP = 'JACOBI'), then subroutine MODE is used to convert SYMX into the linear array (EVAL) form required by the IBM library subroutine EIGEN. The other

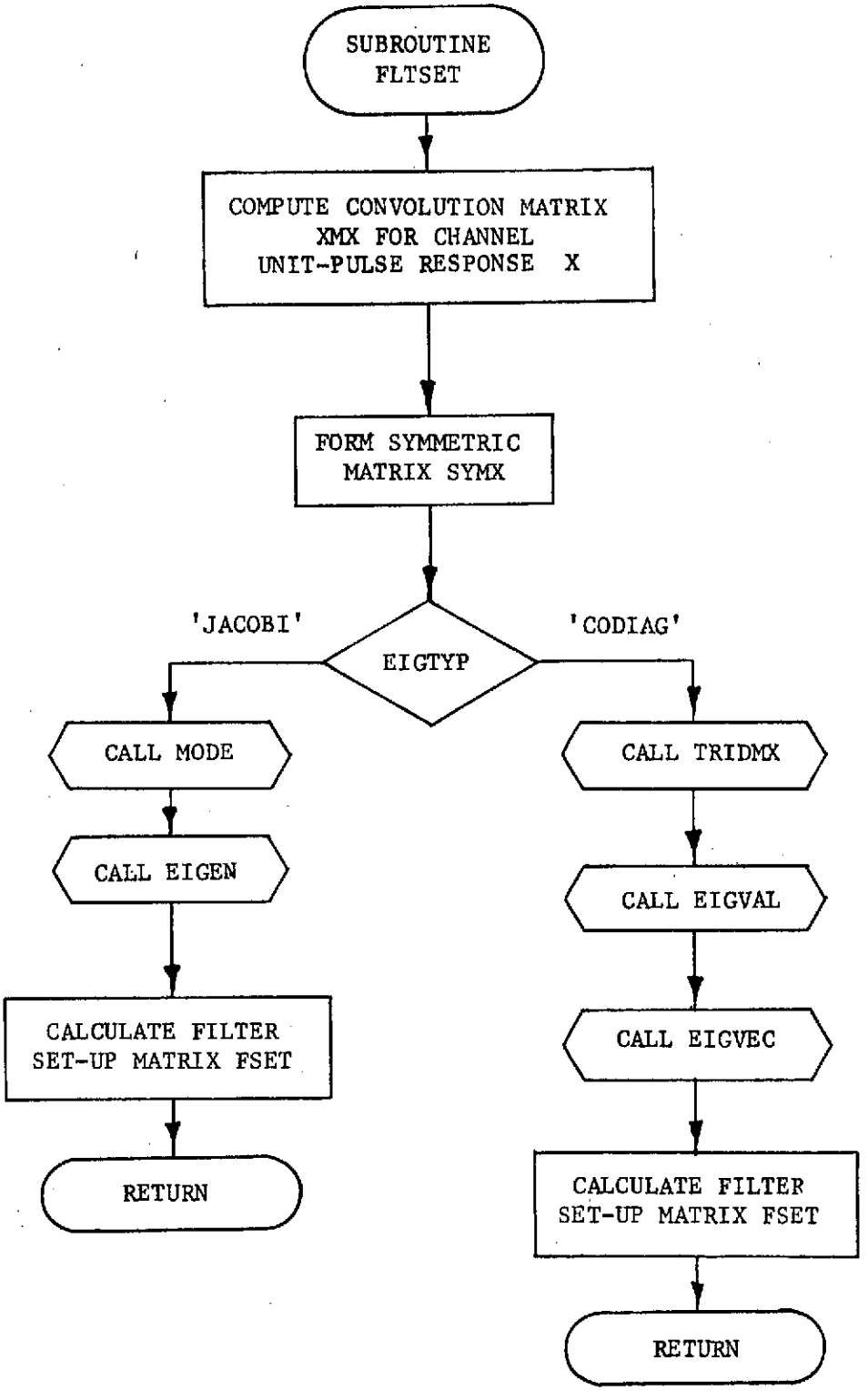

Fig. C.4. Subroutine FLTSET

```
      SUBROUTINE FLTSET (M,N,X,EIGTYP)
      DIMENSION X(32),XMX(32,32), EGVL(32),D(32),SD(32),
     1 TEMP1(32),TEMP2(32),SYMX(32,32),EVAL(528),EVCT(1024)
      COMMON EGVC(32,32),FSET(32,32)
      EQUIVALENCE (SYMX(1,1),FSET(1,1),EVCT(1))
      EQUIVALENCE (EVAL(1),EGVL(1)),(EVAL(33),D(1)),
     1 (EVAL(65),SD(1)),(EVAL(97),TEMPU(I)),(EVAL(129),
     2 TEMP2(1))
C
C     FORM THE CHANNEL OUTPUT MATRIX XMX
      DO 1 I=1,M
      DO 1 J=1,N
      XMX(I,J)=0
      IF (I.GE.J) XMX(I,J)=X(I-J+1)
 1    CONTINUE
C
C     FORM THE REAL SYMMETRIC MATRIX SYMX=(XMX)T*XMX
      DO 2 I=1,N
      DO 2 J=1,N
      SYMX(I,J)=0
      DO 3 K=1,M
 3    SYMX(I,J)=SYMX(I,J)+XMX(K,I)*XMX(K,J)
 2    CONTINUE
C
      IF (EIGTYP.EQ.'JACOBI') GO TO 10
      IF (EIGTYP.EQ.'CODIAG') GO TO 5
C     OTHERWISE PRINT DIAGNOSTIC AND GO TO 10
      WRITE (6,6)
 6    FORMAT (10X,'****EIGNEVALUE METHOD NOT SPECIFIED****')
      GO TO 10
C
 5    CONTINUE
C     COMPUTE THE EIGENVALUES AND EIGENVECTORS OF FSET
      CALL TRIDMX(N,32,SYMX,D,SD)
      CALL EIGVAL(N,EGVL,D,SD,TEMP1,TEMP2)
      CALL EIGVEC(N,32,SYMX,D,SD,EGVL,EGVC,TEMP1,TEMP2)
C
C     CALCULATE THE FILTER SET-UP MATRIX FSET
      DO 4 J=1,N
      EGVL(J)=1/SQRT(EGVL(J))
      DO 4 I=1,N
      FSET(I,J)=EGVL(J)*EGVC(I,J)
 4    CONTINUE
      RETURN
C
 10   CONTINUE
```

Fig. C.5. Listing of Subroutine FLTSET

```
C
      CALL MODE (N,SYMX,EVAL)
      CALL EIGEN (EVAL,EVCT,N,0)
C
C     CALCULATE THE FILTER SET-UP MATRIX
      DO 11 JJ=1,M
      J=M-JJ+1
      IF (J.LE.N) GO TO 12
      DO 13 I=1,N
 13   FSET(I,J)=0
      GO TO 11
 12   K=(J-1)*N
      KK=J*(J+1)/2
      EVAL(KK)=1/SQRT(EVAL(KK))
      DO 14 I=1,N
 14   FSET(I,J)=EVAL(KK)*EVCT(K+I)
 11   CONTINUE
      RETURN
      END
```

Fig. C.5. (Continued) Listing of Subroutine FLTSET

method (EIGTYP = 'CODIAG') is that of Householder, which uses the UNIVAC library subroutines TRIDMX to convert SYMX to tridiagonal form, then EIGVAL and EIGVCT to compute the corresponding eigenvalues (EVAL) and eigenvectors (EVCT). These subroutines require several working matrices (TEMP1, TEMP2, SD, D). After the eigenvalues and eigenvectors are determined, FLTSET computes the filter set-up matrix (FSET) and returns. In the notation of 2.5, the columns of FSET are given by $\mu_k^{-1}\,\xi_k$, $k = 0, 1, \ldots N-1$.

Subroutine MODE places the upper triangular part of a given real symmetric matrix (SYM of dimension DIM) in a linear array (EVAL) columnwise starting with column 1 as required by subroutine EIGEN, then returns. The corresponding listing is given in Fig. C.6.

For any order (M) which is a multiple of four and is not greater than 32, subroutine HADGEN generates a Hadamand matrix (H) by one of three procedures, depending on the order desired. The flow chart is shown in Fig. C.7 and the corresponding listing in Fig. C.8. If M is a power of 2, i.e., 4, 8, 16, and 32, then H is formed in the manner discussed in A.2 using subroutine DOUBLE to repeatedly make the substitutions

$$1 \leftarrow \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \text{and} \qquad -1 \leftarrow \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}.$$

```
      SUBROUTINE MODE(DIM,SYM,LIN)
C     THIS PROCEDURE PLACES THE UPPER TRIANGULAR PART OF
C     SQUARE REAL SYMMETRIC MATRIX SYM IN LINEAR ARRAY LIN
C     IN MODE 1 FORM AS REQUIRED BY SUBROUTINE EIGEN
C
      DIMENSION SYM(32,32),LIN(528)
      INTEGER DIM
      REAL LIN
      L=DIM*(DIM+1)/2
      DO 1 I=1,L
    1 LIN(I)=0
      DO 2 J=1,DIM
      DO 2 I=1,J
      K=J*(J+1)/2
      IF (I.LT.J) K=I+(J-1)*J/2
      LIN(K)=SYM(I,J)
    2 CONTINUE
      RETURN
      END
```

C.6. Listing of Subroutine MODE

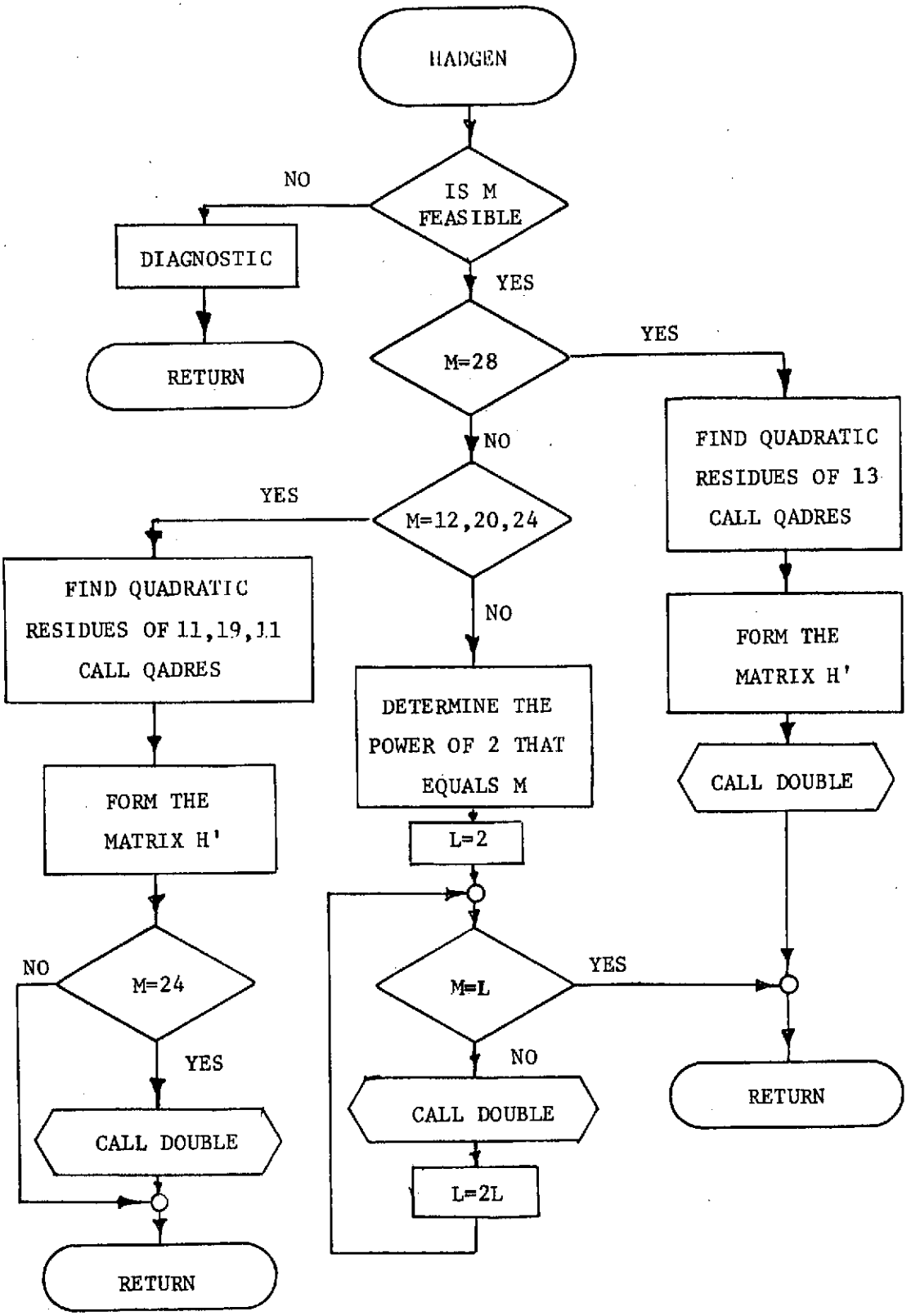

Fig. C.7. Subroutine HADGEN

```
      SUBROUTINE HADGEN (M)
C     THIS PROCEDURE GENERATES A BINARY HADAMARD MATRIX OF
C     ORDER M=4*K WITH K<=8, AND CAN EASILY BE EXTENDED
C     TO K>8
C
      COMMON H(32,32)
      INTEGER RES(20)
      IF((M/4)*4.NE.M.OR.M.GT.32) GO TO 25
C
C     IF M=28 FORM A SIZE 14 MATRIX AND DOUBLE
      IF(M.NE.28) GO TO 10
      LJ=13
      CALL QADRES (LJ,LK,RES)
      LJ=LJ+1
      DO 1 I=1,LJ
      DO 1 J=I,LJ
      IF (J.NE.I) GO TO 2
      H(I,J)=0
      GO TO 1
 2    IF(I.NE.1) GO TO 3
 5    H(I,J)=1
      H(J,I)=1
      GO TO 1
 3    DO 4 K=1,LK
      IF(J-I.EQ.RES(K)) GO TO 5
 4    CONTINUE
      H(I,J)=-1
      H(J,I)=-1
 1    CONTINUE
      CALL DOUBLE (LJ)
      RETURN
C
C     IF M=12 OR M=20 FORM THE MATRIX DIRECTLY
C     IF M=24 FORM A SIZE 12 MATRIX AND DOUBLE
 10   IF(M.NE.12.AND.M.NE.20.AND.M.NE.24) GO TO 20
      LJ=11
      IF(M.EQ.20) LJ=19
      CALL QADRES (LJ,LK,RES)
      LJ=LJ+1
      DO 11 I=1,LJ
      DO 11 J=I,LJ
      IF(I.NE.1) GO TO 13
      H(I,J)=1
      H(J,I)=1
      GO TO 11
 13   IF(J.NE.I) GO TO 15
```

Fig. C.8. Listing of Subroutine HADGEN

```
      H(I,J)=-1
      GO TO 11
 15   DO 16 K=1,LK
      IF(J-1.EQ.RES(K)) GO TO 18
 16   CONTINUE
      H(I,J)=-1
      H(J,I)=1
      GO TO 11
 18   H(I,J)=1
      H(J,1)=-1
 11   CONTINUE
      IF(M.NE.24) GO TO 19
      CALL DOUBLE (LJ)
 19   CONTINUE
      RETURN
C
C     SINCE M=2**K, FORM A SIZE 2 MATRIX AND DOUBLE
C     RECURSIVELY K-1 TIMES
 20   H(1,1)=1
      H(1,2)=1
      H(2,1)=1
      H(2,2)=-1
      LI=1
      IF(M.EQ. 8) LI=2
      IF(M.EQ.16) LI=3
      IF(M.EQ.32) LI=4
      L=2
 22   DO 24 I=1,LI
      CALL DOUBLE (L)
 24   L=2*L
      RETURN
 25   WRITE (6,26)
 26   FORMAT (1H1,'***THE VALUE OF M IS INCORRECT***')
      RETURN
      END
```

Fig. C.8. (Continued) Listing of Subroutine HADGEN

If M is not a power of 2 then the generating procedure is slightly more complicated, and requires the notion of the Legendre symbol. If the congruence relation $r^2 \equiv q \pmod{p}$ has a solution for integers p, q, and r, then q is called a quadratic residue of p, otherwise it is called a quadratic nonresidue. The Legendre symbol (q/p) is defined for a prime p to be 1 when q is a quadratic residue of p, and -1 when q is a quadratic nonresidue. The remaining values of M now fall into two categories. If $M = 2^k(p + 1)$, $k = 0, 1, \ldots$, for some prime $p = 3 \pmod 4$, i.e., M = 12, 20, 24, then the (p + 1) x (p + 1) array H' is first formed, with each element computed according to

$$h_{ij} = \begin{cases} 1 & \text{if } i = 1 \text{ or } j = 1 \\ -1 & \text{if } i = j \text{ and } i > 1 \\ (j-i/p) & \text{otherwise} \end{cases}$$

where the quadratic residues are determined by QUADRES. If M = p + 1, then H is taken to be H', otherwise M = 2(p + 1) and DOUBLE is called to generate H before HADGEN returns. Finally, if $M = 2^k(p + 1)$ for prime $p = 1 \pmod 4$, i.e., M = 28, then H' is computed according to

$$h_{ij} = \begin{cases} 1 & \text{if } i = 1 \text{ or } j = 1 \text{ but } i \neq j \\ 0 & \text{if } i = j \\ (j-i/p) & \text{otherwise,} \end{cases}$$

and again QADRES is called to obtain the quadratic residues. Since M = 2(p + 1), then DOUBLE is called to obtain H before returning. A detailed treatment of the generation of Hadamard matrices for all order up to 100, and many orders greater than 100, can be found in the literature (44,45). Consequently, HADGEN could be expanded to handle considerably larger orders than the present value of 32.

Subroutine DOUBLE doubles the size of a given matrix $H' = [h'_{ij}]$ with elements 1, -1, and 0, to obtain a new matrix by making the following substitutions,

$$h'_{ij} = 1 \leftarrow \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad h'_{ij} = 0 \leftarrow \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}, \text{ and } h'_{ij} = -1 \leftarrow \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}.$$

The original matrix is overwritten by the new matrix H which is binary, i.e., all elements are 1 or -1. The listing is found in C.9, and the algorithm is to simply test each element h_{ij} and make the corresponding matrix substitution just indicated.

Subroutine QADRES computes the quadratic residues of a given prime defined in the discussion of HADGEN, and sorts them in a stack by increasing value. It then returns the stack and the number of elements in the stack. The flowchart and the listing are shown in Figs. C.10 and C.11, respectively.

```
      SUBROUTINE DOUBLE (L)
C     THIS PROCEDURE SUBSTITUTES A PARTICULAR 2X2 MATRIX
C     FOR EACH -1,0,+1 ELEMENT OF MATRIX H , THUS
C     DOUBLING THE SIZE
C
      COMMON V(32,32)
C     SET INDEXING SO THAT NO ORIGINAL ELEMENTS ARE ERASED
      DO 1 I=1,L
      II=L-I+1
      III=2*II
      DO 1 J=1,L
      JJ=L-J+1
      JJJ=2*JJ
C
C     TEST FOR A +1 ELEMENT
      IF(V(II,JJ)-0.01) 3,2,2
 2    V(III,JJJ)=-1
      V(III,JJJ-1)=1
      V(III-1,JJJ)=1
      V(III-1,JJJ-1)=1
      GO TO 1
C
C     TEST FOR A -1 ELEMENT
 3    IF(V(II,JJ)+0.01) 4,4,5
 4    V(III,JJJ)=1
      V(III,JJJ-1)=-1
      V(III-1,JJJ)=-1
      V(III-1,JJJ-1)=-1
      GO TO 1
C
C     ASSUME A 0 ELEMENT BY DEFAULT
 5    V(III,JJJ)=-1
      V(III,JJJ-1)=-1
      V(III-1,JJJ)=-1
      V(III-1,JJJ-1)=1
 1    CONTINUE
C
      RETURN
      END
```

Fig. C.9. Listing of Subroutine DOUBLE.

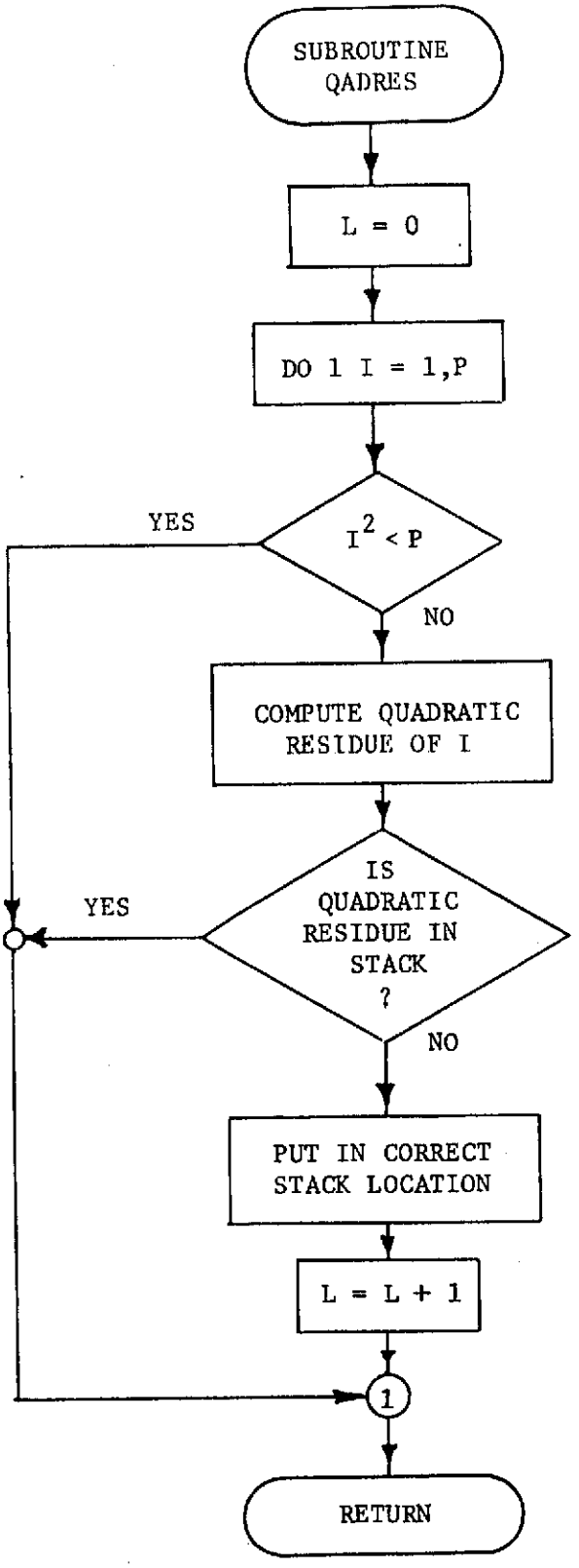

Fig. C.10. Subroutine QADRES

```
      SUBROUTINE QADRES (P,L,RES)
C     THIS PROCEDURE COMPUTES THE QUADRATIC RESIDUES
C     OF ANY PRIME P AND STORES THEM IN STACK RES(L)
C
      INTEGER P,QRES,PM,RES(20)
      L=0
      PM=P-1
      DO 1 I=1,PM
      ISQ=I*I
      IF(ISQ.LT.P) GO TO 1
C     COMPUTE THE QUADRATIC RESIDUE FOR EACH INTEGER  I<P
      QRES=ISQ-(ISQ/P)*P
      IF(L.GT.0) GO TO 4
      RES(1)=QRES
      GO TO 3
 4    CONTINUE
C     CHECK THE STACK RES FOR  DUPLICATION
      DO 5 J=1,L
      IF(RES(J).EQ.QRES) GO TO 1
 5    CONTINUE
C     ADD THE NEW QUADRATIC RESIDUE TO THE STACK AND
C     SHIFT POINTER L
      RES(L+1)=QRES
      DO 2 J=1,L
      IF(QRES.GT.RES(L-J+1)) GO TO 3
      RES(L-J+2)=RES(L-J+1)
 2    RES(L-J+1)=QRES
 3    L=L+1
 1    CONTINUE
      RETURN
      END
```

Fig. C.11 Listing of Subroutine QADRES

C.3 COMPUTATION OF EIGENVALUES AND EIGENVECTORS

The computer simulation described in this appendix uses one of two methods to obtain the eigenvalues and eigenvectors of an N x N real symmetric matrix $\underline{A}$. Both methods are iterative, thus adaptable for machine computation, and both make use of a series of unitary similarity transformations to reduce the given matrix to a simplified form.

The first method [54], originally proposed by Jacobi in 1864, has been revised in recent years for application to modern digital computers. The original method is based on applying an infinite series of particular similarity transformations of the form $\underline{A}_{i+1} = \underline{S}_i^T \underline{A}_i \underline{S}_i$ with $\underline{A}_0 = \underline{A}$. The matrices $\underline{S}_i$ are plane (two-dimensional) rotations, and by properly selecting the plane and angle of rotation, any particular off-diagonal element can be annihilated, i.e., reduced to zero. Although the annihilation of other elements will in general cause previously annihilated elements to be nonzero, Jacobi showed that all off-diagonal elements tend to zero in the limit. By defining $\underline{S}$ and $\underline{D}$ to be the limits of $\underline{S}_i$ and $\underline{D}_i$, respectively, it follows that

$$\underline{D} = \underline{S}^T \underline{A} \underline{S} ,$$

with $\underline{D}$ diagonal and $\underline{S}$ unitary. This implies that the diagonal elements of $\underline{D}$ are the eigenvalues of $\underline{A}$, with the corresponding eigenvectors being the columns of $\underline{S}$.

With machine computations the process is, of course, terminated after a finite number of steps as soon as the magnitude of the largest off-diagonal element becomes less than some predetermined threshold value. The convergence rate can also be improved by properly selecting the order in which the elements are annihilated.

The second method [55] uses a procedure suggested by Householder to convert a real symmetric matrix into tridiagonal (codiagonal) form $\underline{C}$ with a series of N - 2 unitary transformations of the form $\underline{I} - 2\,\underline{u}\,\underline{u}^T$ where $||\underline{u}||=1$. By properly selecting $\underline{u}$, any column and the corresponding row can be transformed to contain only two elements, thus giving the tridiagonal form. Once $\underline{A}$ is in tridiagonal form the eigenvalues of $\underline{C}$, hence of $\underline{A}$, are found by noticing that the set of characteristic polynomials corresponding to the leading principle submatrices forms a Sturm sequence. Each root of the largest polynomial is an eigenvalue of $\underline{A}$, and the roots are found by obtaining successively smaller intervals in which each lies. The eigenvectors are then found for $\underline{C}$ by applying the inverse power method [56] which is particularly simple for codiagonal matrices. These eigenvectors are then transformed into those of $\underline{A}$ by using the same series of unitary transformations obtained in computing $\underline{C}$ initially.

LIST OF REFERENCES

1. J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, New York: John Wiley and Sons, 1965.

2. L. R. Rabiner, et al., "Terminology in digital signal processing," IEEE Trans. Audio and Electroacoustics, vol. AU-20, pp. 322-327, Dec. 1972.

3. R. W. Lucky, "Techniques for adaptive equalization of digital communication systems," Bell Syst. Tech. J., vol. 45, pp. 255-286, Feb. 1966.

4. R. W. Lucky, "Automatic equalization for digital communication," Bell Syst. Tech. J., vol. 44, pp. 547-588, April, 1965.

5. B. Widrow, "Adaptive filters 1:fundamentals," Stanford Electronics Lab, Stanford, Calif., Report SEL-60-162, Dec. 1966.

6. N. Wiener, Extrapolation, Interpolation, and Smoothing of Time Series, New York: John Wiley and Sons, 1957.

7. R. W. Lucky and H. R. Rudin, "An automatic equalizer for general-purpose communication channels," Bell Syst. Tech. J., vol. 46, pp. 2179-2208, Nov. 1967.

8. A. Gersho, "Adaptive equalization of highly dispersive channels for data transmission," Bell Syst. Tech. J., vol. 48, pp. 55-70, Jan. 1969.

9. A. Lender, "Decision-directed digital adaptive equalization technique for high-speed data transmission," IEEE Trans. Communication Technology, vol. COM-18, pp. 625-632, Oct. 1970.

10. D. Hirsch and W. J. Wolf, "A simple adaptive equalizer for efficient data transmission," IEEE Trans. Communication Technology, vol. COM-18, pp. 5-11, Feb. 1970.

11. A. Gersho, "Linear adaptation," Proc. of the Symposium of Computer Processing in Communications, vol. 19, pp. 653-664, Apr. 1969.

12. R. W. Lucky, J. Salz, and E. J. Weldon, Jr., Principles of Data Communication, New York: McGraw-Hill, 1968, Chaps. 5-6.

13. R. D. Gitlin, J. E. Mazo, and M. G. Taylor, "On the design of gradient algorithms for digitally implemented adaptive filters," IEEE Trans. Circuit Theory, vol. CT-20, pp. 125-136, Mar. 1973.

14. J. G. Prokias, "Adaptive digital filters for equalization of telephone channels," IEEE Trans. Audio and Electroacoustics, vol. AU-18, pp. 195-200, Jun. 1970.

15. G. D. Forney, Jr., "The Viterbi algorithm," Proc. IEEE, vol. 61, pp. 268-278, Mar. 1973.

16. F. R. Magee, Jr. and J. G. Prokias, "Adaptive maximum-liklihood sequence estimation for digital signaling in the presence of intersymbol interference," IEEE Trans. Information Theory, vol. 19, pp. 120-124, Jan. 1973.

17. S. H. Richman and M. Schwartz, "Dynamic programming training period for an MSE adaptive equalizer," IEEE Trans. Communications, vol. COM-20, pp. 857-864, Oct. 1972.

18. R. E. Lawrence and H. Kaufman, "The Kalman filter for the equalization of a digital communications channel," IEEE Trans. Communication Technology, vol. COM-19, pp. 1137-1141, Dec. 1971.

19. D. A. George, et al., "Channel equalization for data transmission," Engineering J., vol. 53, pp. 20-31, May 1970.

20. G. K. McAuliffe, "Practical adaptive equalizers for data transmission," Proc. Western Electronics Show and Convention, vol. 13, pp. 11/3.1-11/3.5, Aug. 1969.

21. T. J. Schonfeld and M. Schwartz, "Rapidly converging second-order tracking algorithms for adaptive equalization," IEEE Trans. Information Theory, vol. IT-17, pp. 572-579, Sep. 1971.

22. R. D. Gitlin and J. E. Mazo, "Comparison of some cost functions for automatic equalization," IEEE Trans. Communications, vol. COM-21, pp. 233-237, Mar. 1973.

23. A. Gersho, "Automatic time-domain equalization with transversal filters," Proc. National Electronics Conference, vol. 22, pp. 428-432, Oct. 1966.

24. T. J. Schonfeld and M. Schwartz, "A rapidly converging training algorithm for an adaptive equalizer," IEEE Trans. Information Theory, vol. IT-17, pp. 431-439, Jul. 1971.

25. R. W. Chang, "A new equalizer structure for fast start-up digital communication," Bell Syst. Tech. J., vol. 50, pp. 1969-2014, Jul-Aug. 1971.

26. R. Ramachandran and W. Steenaart, "Optimal equalization of discrete signals passed through a random channel," Rennsselaer Polytechnic Institute Project Themis Report DAAB07-C-0365, DDC AD-753873, 1972.

27. S. W. Golomb, Digital Communications with Space Applications, Englewood Cliffs, N.J.: Prentice-Hall, 1964, chaps. 7-8.

28. J. J. Stiffler, Theory of Synchronous Communications, Englewood Cliffs, N. J.: Prentice Hall, 1971, chaps. 6-9.

29. W. Zangwill, Nonlinear Programming: A Unified Approach, Englewood Cliffs, N.J.: Prentice-Hall, 1969, p. 31.

30. O. I. Elgerd, Control Systems Theory, New York: McGraw-Hill, 1967, p. 182.

31. D. W. Burlage and R. C. Houts, "Applications of linear programming to the time-domain design of digital-filter equalizers," IEEE Trans. Communications, vol. COM-21, pp. 1417-1422, Dec. 1973.

32. Lucky, Saltz, and Weldon, op. cit., pp. 59-75.

33. Instructions Model 300 and Model 302 Variable Electronic Filters, Spencer-Kennedy Laboratories, Inc., Boston, Jan. 1964.

34. R. J. Wescott, "An experimental adaptively equalized modem for data transmission over the switched telephone network," Proc. Conference on Digital Processing of Signals, pp. 195-212, Apr. 1972.

35. C. A. Desoer, Notes for a Second Course on Linear Systems, New York: Van Nostrand Reinhold, 1970, pp. 2-3.

36. Ibid., p. 5.

37. S. K. Berberian, Introduction to Hilbert Space, New York: Oxford University Press, 1961, p. 85.

38. F. Ayers, Jr., Theory and Problems of Modern Algebra, New York: McGraw-Hill, 1965, p. 167.

39. Desoer, op. cit., p. 15.

40. B. Noble, Applied Linear Algebra, Englewood Cliffs, N. J.: Prentice Hall, 1969, p. 321.

41. Ibid., p. 333.

42. N. Dunford and J. T. Schwartz, Linear Operators, Part III, New York: Wiley-Interscience, 1971.

43. Berberian, op. cit., p. 163.

44. Golomb, op. cit., pp. 35-56.

45. R. E. A. C. Paley, "On orthogonal matrices," Journal of Mathematics and Physics, vol. 12, 1933, pp. 311-20.

46. D. M. Wiberg, Theory and Problems of State Space and Linear Systems, New York: McGraw-Hill, 1971, p. 84.

47. Ibid., p. 85.

48. Ibid., p. 86.

49. E. I. Jury, Theory and Application of the z-Transform Method, New York: John Wiley and Sons, 1964.

50. B. Gold and C. M. Rader, Digital Processing of Signals, New York: McGraw-Hill, 1969, chaps. 2-3.

51. L. J. Griffiths, "A simple adaptive algorithm for real-time processing in antenna arrays," Proc. IEEE, vol. 57, pp. 1696-1704, Oct. 1969.

52. System/360 Scientific Subroutine Package, Version III, Programmer's Manual (360A-CM-03X), fifth ed., International Business Machines Corp., 1968, pp. 164-165.

53. UNIVAC Large Scale Systems Math-Pack Programmer's Reference, Sperry Rand Corp., 1970, chap. 9.

54. J. Greenstadt, "The determination of the characteristic roots of a matrix by the Jacobi method," in Mathematical Methods for Digital Computers, vol. I, A. Ralston and H. S. Wilf, Eds., New York: John Wiley and Sons, 1962, pp. 84-91.

55. J. Ortega, "The Givens-Householder method for symmetric matrices," in Mathematical Methods for Digital Computers, vol. II, A. Ralston and H. S. Wilf, Eds., New York: John Wiley and Sons, 1967, pp. 94-115.

56. J. H. Wilkinson, "The calculation of eigenvectors of codiagonal matrices," Computer J., vol. 1, pp. 148-152, Jul. 1958.

COMMUNICATION SYSTEMS GROUP

RECENT REPORTS

A Digital Technique to Compensate for Time-Base Error in Magnetic Tape Recording, R. S. Simpson, R. C. Houts and D. W. Burlage, August 1968.

A Study of Major Coding Techniques for Digital Communication, R. S. Simpson and J. B. Cain, January, 1969.

Study of Correlation Receiver Performances in the Presence of Impulse Noise, R. C. Houts and J. D. Moore, February, 1971.

Computer-Aided Design of Digital Filters, R. C. Houts and D. W. Burlage, March, 1971.

Characterization of Impulse Noise and Analysis of Its Effects Upon Correlation Receivers, R. C. Houts and J. D. Moore, October, 1971.

The Use of Linear Programming Techniques to Design Optimal Digital Filters for Pulse Shaping and Channel Equalization, R. C. Houts and D. W. Burlage, April, 1972.

A Comparative Analysis of Digital Baseband Signals, R. C. Houts and T. A. Green, June, 1972.

Design of FIR Digital Filters for Pulse Shaping and Channel Equalization Using Time-Domain Optimization, R. C. Houts and G. L. Vaughn, May, 1974.